

Řízení robotického ramene pomocí programovatelného automatu

Control of Robotic Arm using Programmable Logic Controller

Bc. Michal Duřák

Diplomová práce

Vedoucí práce: prof. Ing. Jiří Koziorek, Ph.D.

Ostrava, 2021

Abstrakt

Cílem diplomové práce je rozebrat různé možnosti řízení robotu. Následně popsat analýzu řízení robotu pomocí softwarových modulů, jež představuje knihovna mxAutomation. Provést definování a popis vybrané laboratorní robotické úlohy, kterou reprezentuje výuková buňka ready2_educate s KUKA robotem KR3 R540. Navrhnout a vytvořit program pro PLC, jenž bude řídit pohyby šestiosého robotického ramene. Vytvořit vizualizaci, která bude ovládaná z HMI panelu. Zhodnotit dosažené výsledky a porovnat zvolenou metodu s metodou klasickou pro tvorbu robotických aplikací.

Klíčová slova

mxAutomation, KUKA, PLC, SIMATIC, Siemens, KR3 R540, ready2_educate, KR C4 compact

Abstract

The aim of the diploma thesis is to analyze various possibilities of robot control. Then describe the analysis of robot control using software modules, which is represented by the mxAutomation library. Define and describe the selected laboratory robotic task, which is represented by the training cell ready2_educate with the KUKA robot KR3 R540. Design and create a program for a PLC that will control the movements of the six-axis robotic arm. Create a visualization that will be controlled from the HMI panel. Evaluate the achieved results and compare the selected method with the classical method for creating robotic applications.

Keywords

mxAutomation, KUKA, PLC, SIMATIC, Siemens, KR3 R540, ready2_educate, KR C4 compact

Poděkování

Rád bych na tomto místě poděkoval vedoucímu práce prof. Ing. Jiří Kozíorkovi, Ph.D za odborné vedení, pomoc a rady při zpracování mé diplomové práce. Dále celé své rodině, která mě podporovala a poskytla mi prostor pro realizaci této práce.

Obsah

Seznam použitých symbolů a zkratk	6
Seznam obrázků	7
Seznam tabulek	10
1 Úvod	11
2 Analýza možností řízení šestiosých stacionárních robotických systémů	13
2.1 Struktura šestiosého průmyslového stacionárního robotu	14
2.2 Způsoby realizace řízení pohybu	16
2.3 Pohybové příkazy u průmyslového robotu KUKA	18
2.4 Aproximace pohybových příkazů	20
2.5 Typy souřadnicových systémů robotu	23
2.6 Hardwarové možnosti řízení stacionárních šestiosých průmyslových robotů	24
2.7 Možnosti programování stacionárních šestiosých průmyslových robotů	25
3 Analýza a popis problematiky ovládání robotu pomocí softwarových modulů	29
3.1 Popis systému a vytváření programu v mxAutomation a jeho zpracování	29
3.2 Členění knihovny mxAutomation	30
3.3 Integrace robotu do hardwarové konfigurace projektu TIA Portal	30
3.4 Import knihovny robotických bloků mxAutomation a použití v programu	34
3.5 Základní členění programu	35
3.6 Typy funkcí a funkčních bloků pro ovládání robotu	35
3.7 Funkční blok KRC_ReadAxisGroup	36
3.8 Funkční blok KRC_WriteAxisGroup	36
3.9 Funkční blok KRC_Initialize	37
3.10 Funkční blok KRC_AutomaticExternal	38
3.11 Funkční blok KRC_AutoStart	39
3.12 Funkční blok KRC_SetOverride	39

3.13	Funkční blok KRC_ReadActualPosition	39
3.14	Funkční blok KRC_ReadActualAxisPos	40
3.15	Funkční blok KRC_Jog	40
3.16	Funkční blok KRC_TouchUP	41
3.17	Funkční blok KRC_Interrupt	41
3.18	Funkční blok KRC_Continue	43
3.19	Funkční blok mxA_ValuesToCOORDSYS	43
3.20	Funkční blok mx_AValuesToAPO	44
4	Definování a popis vybrané laboratorní úlohy	47
4.1	Jednotlivé komponenty robotického stanoviště	48
4.2	Definování robotické úlohy	57
5	Návrh a realizace řídicího programu pro průmyslového robotu	61
5.1	Tvorba hardwarové konfigurace	61
5.2	Členění programu	61
5.3	Hlavní systémové bloky	62
5.4	Funkční bloky pro ovládaní robotu	63
5.5	Programové bloky	65
6	Realizace uživatelské HMI aplikace	74
6.1	Přihlašovací okno HMI aplikace	74
6.2	Okno HMI aplikace pro ovládání robotu	74
6.3	Okno HMI aplikace pro spouštění robotické operace	74
7	Zhodnocení výsledků řešení	77
8	Závěr	79
	Literatura	81
	Přílohy	82
A	Soubory	83
B	Velké obrázky a tabulky	84

Seznam použitých zkratek a symbolů

DOF	– Degrees of freedom (Počet stupňů volnosti)
OLP	– Offline Programming (off-line programování)
PLC	– Programmable Logic Controller (Programovatelný logický automat)
KRL	– Kuka Robot language (Kuka programovací jazyk)
TCP	– Tool Center Point (Nástrojová osa robotu)
RRR	– R-R-R configuration, 3 revolute joints (Rotační Rotační Rotační)
PTP	– Point to point (Z bodu do bodu)
LIN	– Linear motion (Lineární pohyb)
CP	– Continuous path (Kontinuální cesta)
MP	– Many points (Mnoho bodové)
CIRC	– Circular motions (Kruhový pohyb)
PC	– Personal computer (Osobní počítač)
IPC	– Industrial personal computer (Kruhový pohyb)
CNC	– Computer Numerical Control (Počítačové číselné řízení)
FB	– Function block (Funkční blok)
FC	– Function control (Funkce)
DB	– Data block (Datový blok)
HMI	– Human Machine Interface (Rozhraní člověk stroj)

Seznam obrázků

2.1	Průmyslový robot	13
2.2	Struktura robotu	14
2.3	Blokové schéma řídicího systému robotu	15
2.4	Popis konstrukce robotu	16
2.5	Ukázka programovacího/ovládacího panelu	17
2.6	Řízení pohybu metodou PTP	17
2.7	Řízení pohybu typu CP	18
2.8	Řízení pohybu typu MP	19
2.9	Pohybový příkaz PTP	19
2.10	Pohybový příkaz CIRC	20
2.11	Pohybový příkaz LIN	21
2.12	Aproximace pohybového příkazu PTP	22
2.13	Aproximace pohybového příkazu LIN	22
2.14	Aproximace pohybového příkazu CIRC	23
2.15	Souřadnicové systémy World, Tool, Base, Robroot	24
2.16	Hardwarové možnosti řízení robotu	25
2.17	Nástroje pro on-line programování (PC,Pendant)	26
2.18	Znázornění postupu při on-line programování	26
2.19	Znázornění simulačního prostředí pro off-line programování	27
2.20	Znázornění postupu při off-line programování	27
3.1	Výběr a instalace souboru GSDML „KUKA-KRC4-ProfiNet“	31
3.2	Přidání řídicí jednotky robotu do hardware konfigurace	33
3.3	Datový blok pro pět robotů v datovém bloku MaxDB Robots	33
3.4	Vytvoření komunikace mezi robotem a PLC	34
3.5	Globální knihovna funkčních bloků	35
3.6	Funkční blok pro čtení dat z robotu	36
3.7	Funkční blok pro zápis dat do robotu	36
3.8	Funkční blok pro inicializaci rozhraní KUKA.PLC mxAutomation	37

3.9	Funkční blok pro navázání spojení mezi PLC a robotem kuka	38
3.10	Funkční blok pro uspořádání signálu	39
3.11	Funkční blok pro nastavení rychlosti robotu	39
3.12	Funkční blok pro čtení kartézských souřadnic nástrojové osy robotu	40
3.13	Funkční blok pro čtení pozic natočení jednotlivých os robotu	41
3.15	Znázornění funkčního bloku KRC TouchUP	41
3.14	Znázornění funkčního bloku KRC Jog	42
3.16	Znázornění funkčního bloku pro přerušení vykonávání programu	43
3.17	Znázornění funkčního bloku pro pokračování programu po přerušení	43
3.18	Znázornění funkčního bloku, který umožňuje výběr souřadného systému robotu . . .	43
3.19	Znázornění funkčního bloku, který definuje typ pohybové funkce	44
3.20	Znázornění trajektorie robotu	45
4.1	Robotické pracoviště KUKA ready2_educate	47
4.2	Řídicí jednotka KR C4 compact	48
4.3	Ovládací a zobrazovací prvky	49
4.4	KUKA smartPAD přední strana	51
4.5	KUKA smartPAD zadní strana	51
4.6	SIMATIC HMI panel KTP 900 Basic	52
4.7	Chapadlo robotu	52
4.8	Montážní deska pneumatiky	54
4.9	Montážní deska elektriky	54
4.10	PLC SIMATIC ET 200SP	55
4.11	Výukové přípravky robotického pracoviště KUKA ready2 educate	55
4.12	Rozsah pracovní zóny robotu KUKA KR 3 R540	57
4.13	Výuková robotická buňka	59
4.14	Prvky robotické buňky použité pro vytvoření robotické aplikace	60
5.1	HW konfigurace PLC SIMATIC ET 200SP	62
5.2	Síťová HW konfigurace celého projektu	62
5.3	Hlavní blok, který obsahuje celý program	62
5.4	Funkční bloky pro čtení a zápis dat do robotické jednotky	63
5.5	Funkční blok KukuControlVSB	64
5.6	Funkční blok pro zapnutí rozhraní mxAutomation	64
5.7	Funkční blok pro ovládaní ofuku a přísavky	65
5.8	Funkční blok pro ovládaní chapače	65
5.9	Funkční blok PickPlaceCubeVSB	66
5.10	Funkční blok MoveAxisAbsoluteVSB	69
5.11	Funkční blok MoveLinearAbsoluteVSB	70

5.12	Funkční blok ConnectSuctionCup	70
5.13	Funkční blok OperationPickCube	71
5.14	Funkční blok PlaceCube	72
5.15	Funkční blok PickCubeFromPosition	72
5.16	Funkční blok InsertingCubeIntoStorage	73
5.17	Funkční blok DisconnectSuctionCup	73
6.1	Přihlašovací okno HMI aplikace	75
6.2	Okno HMI aplikace pro ovládání robotu	75
6.3	Okno HMI aplikace pro spouštění robotické operace	76
7.1	Inline formulář pro KRL	77
7.2	Funkční blok knihovny mxAutomation	78
B.1	Diagram zpracování programu	84
B.2	Aktivitní diagram návrhu programu	85

Seznam tabulek

3.1	Různé verze PROFINETU s přidruženým souborem GSDML [20]	32
3.2	Popis důležitých vstupních hodnot bloku KRC_AutomaticExternal [20]	38
3.3	Popis důležitých vstupních hodnot bloku mxA_ValuesToCOORDSYS [20]	44
3.4	Popis důležitých vstupních hodnot bloku mxA_ValuesToAPO [20]	46
4.1	Popis k jednotlivým ovládacím prvků robotické stanice [22]	49
4.2	Popis k jednotlivým prvkům KUKA smartPAD [22]	50
4.3	Popis chapadla robotu [22]	53
4.4	Popis montážní desky pneumatiky [22]	53
4.5	Popis montážní desky elektriky [22]	53
4.6	Popis aplikačních prvků [22]	56
4.7	Popis důležitých parametrů KUKA robotu [22]	56
5.1	Vstupy a výstupy funkčního bloku MoveAxisAbsoluteVSB	67
5.2	Vstupy a výstupy funkčního bloku MoveLinearAbsoluteVSB	68

Kapitola 1

Úvod

Tato diplomová práce se zabývá možnostmi řízení šestiosého robotu KUKA pomocí programovatelného automatu od společnosti Siemens. Tato metoda je vcelku nová a není moc materiálu, z něhož by se dala čerpat inspirace pro tvorbu aplikací. K dispozici jsou pouze materiály od společnosti KUKA a Siemens.

Pro tvorbu řídicího programu byla použita knihovna funkcí mxAutomation, která obsahuje všechny důležité funkce a funkční bloky pro ovládání robotu a jeho externích zařízení, navázání komunikace s řídicí jednotkou robotu a programovatelným automatem. Celá úloha je vytvořena pro robotickou buňku ready2_educate, jejíž součástí jsou další prvky, jež jsou použity při tvorbě programu.

První část práce se zabývá analýzou možností řízení robotu, věnuje se rozboru všech částí, z nichž je robot složen a které jsou důležité pro jeho řízení. Z pohledu algoritmů řízení jsou následně popsány různé druhy pohybu, jaké robot může vykonávat. Poslední částí této kapitoly je analýza řízení robotu z pohledu HW a metod tvorby robotického programu.

V následující části, v níž je problematika řízení robotu přes PLC analyzována za pomoci knihoven mxAutomation, je popsáno, jakým způsobem PLC program zpracovává a rovněž jak ho posílá do řídicí jednotky robotu. Dále jsou zde objasněny důležité funkční bloky pro obsluhu robotu. Následně v souvislosti s návrhem robotické aplikace je provedena důkladná analýza robotické buňky na jednotlivé komponenty, a to od elektroniky a zobrazovacích zařízení přes konstrukční části a výukové prvky až po návrh robotického programu. Na komplexní rozebrání a vysvětlení robotické buňky navazuje definování robotické úlohy, respektive programu, který má robot vykonávat.

V diplomové práci je zpracována i tvorba řídicího programu, jenž byl vytvořen v prostředí TIA Portal za pomoci knihoven mxAutomation. Nejdříve jsou rozebrány funkční bloky pro ovládání robotu a pak se práce soustředí na ty, které vykonávají program. Jsou zde charakterizovány nově vytvořené pohybové bloky, jež se skládají z bloků z knihovny mxAutomation.

V předposlední kapitole je obsažen popis grafické části úlohy, která je realizována pomocí HMI panelu, na němž je vytvořeno grafické rozhraní pro ovládání programu. Vizualizace se skládá ze

tří oken, a to z okna pro přihlašování uživatele, okna pro ovládání robotu a okna pro spouštění programu. Poslední část práce se věnuje zhodnocení a porovnání dvou metod programování robotu jednou z těchto metod je externí řízení robotu přes PLC program a druhou je klasický postup tvorby robotického programu za pomoci inline formulářů.

Kapitola 2

Analýza možností řízení šestiosých stacionárních robotických systémů

Využití průmyslových šestiosých robotických systémů stále roste. Jedním z důvodů je modularita jejich standardizovaných mechanik, která je použitelná v mnoha aplikacích. Jejich výhodou je taky jednoduché uživatelské ovládání, které je přizpůsobeno pro uživatele, kteří nejsou programátoři. Tito roboti stále častěji nahrazují klasické mechaniky právě díky tomu, že se mohou v prostoru dostat do jakéhokoli bodu a také na rozdíl od kinematik lineárních nabízejí možnosti opětovného použití v úplně jiné aplikaci. Na obr. 2.1 lze vidět, jak šestiosý průmyslový robot vypadá. [2][3]



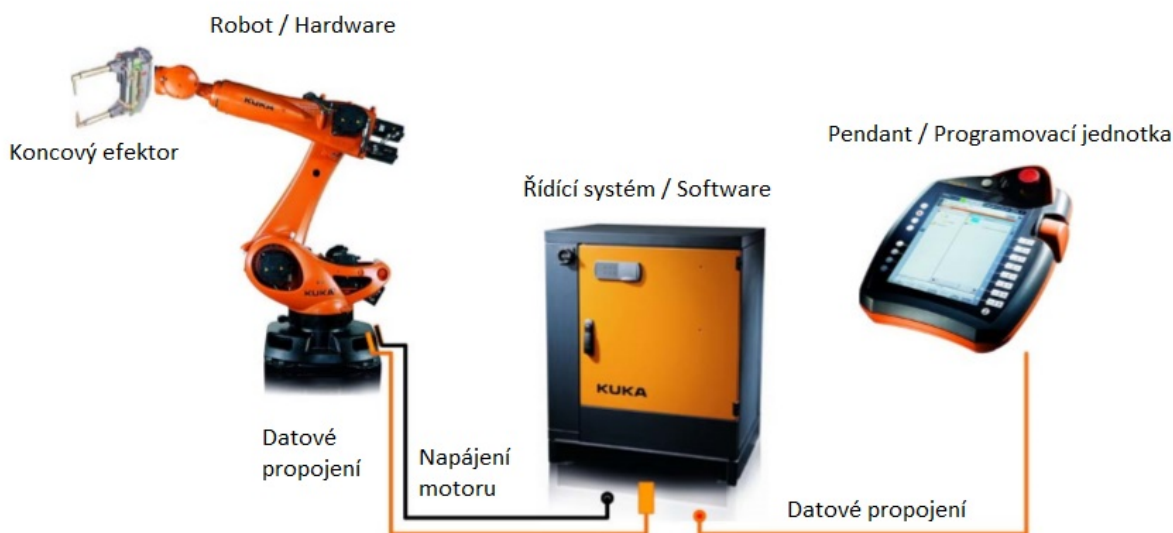
Obrázek 2.1: Průmyslový robot
[1]

Stacionární šestiosý robot je systém se šesti stupni volnosti tzv. 6DOF, jenž je rozšířením základního kinematického řetězce RRR, který je nedostačující, protože nabízí pouze nastavení libovolné

pozice v prostoru, což je mnohdy problém s ohledem na to, že ve spoustě aplikací je vyžadována změna uchopení předmětu. Proto se často tento model nahrazuje právě systémem se šesti stupni volnosti. Jedná se o zařízení využívající sériovou kinematiku s univerzálním použitím. Jeho použití není omezeno pouze na manipulaci, ale je často využíváno na aplikace, jako je svařování, lepení, vrtání, montáž a mnoho dalších. Soustrojí robotu je složeno pouze z vazeb angulárních, jež se mohou pohybovat nezávisle na sobě. Z toho vyplývá jeho největší výhoda, a to vysoká variabilita použití. Na druhou stranu robot s touto kinematikou má i své nevýhody, jako je nižší rychlost, zrychlení a taky méně přesné polohování, které je dané vyšším množstvím vůlí převodovek. [4]

2.1 Struktura šestiosého průmyslového stacionárního robotu

Strukturu všech vestavěných průmyslových systémů lze rozdělit do tří částí, na řídicí, mechanickou a programovací, viz obr. 2.2. Každá z těchto částí je nepostradatelná pro funkčnost robotu. [4] [3]



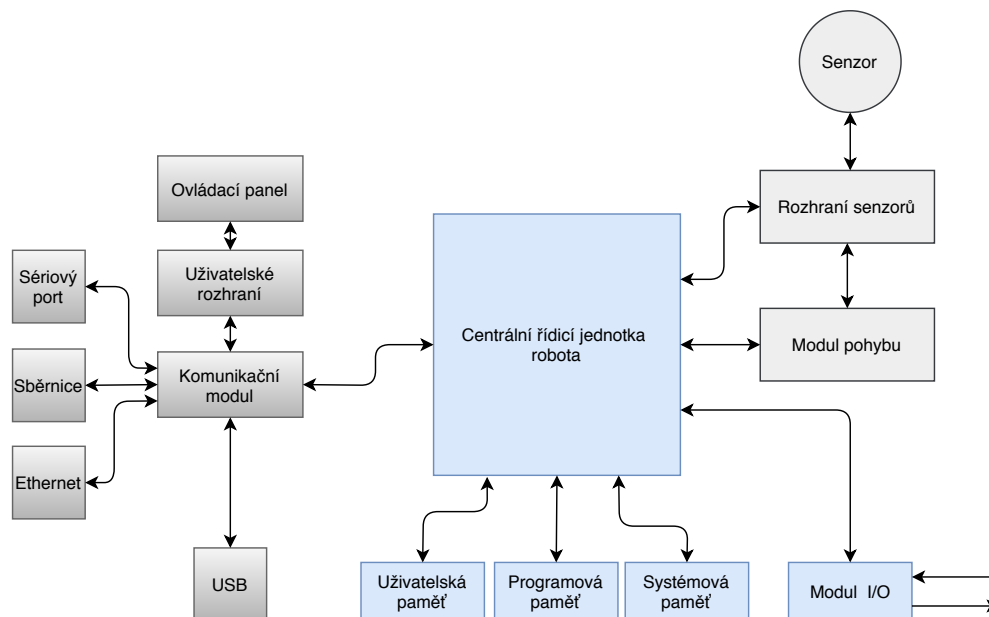
Obrázek 2.2: Struktura robotu
[4]

2.1.1 Řídicí část

Úlohou řídicí části je vyhodnocovat data uložená v paměti řídicího systému průmyslového počítače na základě informací získaných ze snímačů. Na základě těchto dat pak řídicí systém určuje, jaké operace budou vykonány tak, aby byl proveden daný proces. Blokové schéma řídicího systému se nachází na obr. 2.3. [3] [4]

Řídicí systém má v sobě integrované funkce pro řízení polohování a kromě toho také nabízí možnost řízení periferních zařízení přes I/O moduly. Systém jako takový je realizován jako mikroprocesorový počítač, jenž pracuje podle multitaskingového procesu. Několik sekvenční řídicích procesů je možné zpracovávat najednou. [4]

Vstupně výstupní úroveň řídicího systému je dle provedení prezentována jako technologie průmyslových sběrnic nebo jako diskretní I/O. Sériové komunikační rozhraní je možné překonfigurovat a lze ho použít například pro připojení smart zařízení, jako je čtečka čárových kódů nebo kamera či skener pro zpracování obrazu. [4]



Obrázek 2.3: Blokové schéma řídicího systému robota

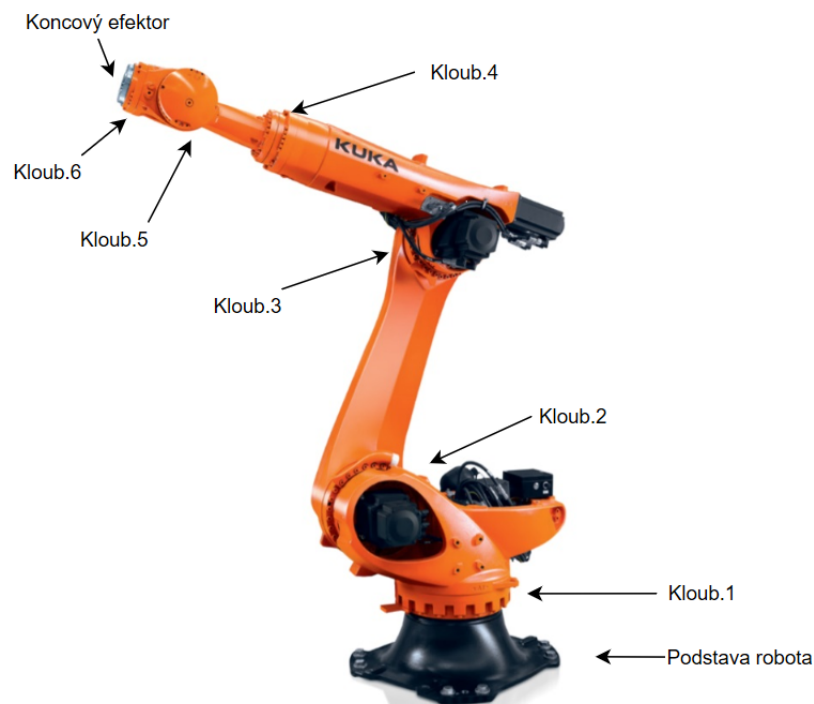
[4]

2.1.2 Mechanická část

Mechanická neboli strojní část průmyslového robota se skládá z kloubů, což jsou jinými slovy pohyblivé spoje. Pomocí kloubu robota je realizován pohyb, vazby mezi nimi tvoří tuhá tělesa. Každý kloub robota poskytuje stupeň volnosti. Mechanická část robota je složena z podstavy, karuselu a jednotlivých článků, které spojují klouby. Koncový efektor je samostatná část robotického zařízení, sloužící k uchycení objektu, s nímž se manipuluje chapadlo nebo taky to může být svařovací pistole. Na obrázku viz 2.4 je zobrazen popis kostry robota. [4]

2.1.3 Programovací a ovládací část

Nastavování a programování robota je realizováno pomocí programovacího panelu tzv. pendantu. Tento nástroj je vždy vybaven velkým displejem a emergenci stop tlačítkem. Na displeji panelu jsou



Obrázek 2.4: Popis konstrukce robotu
[5]

zobrazovány veškeré možnosti nastavení, a krom toho jsou na panelu také indikovány aktuální stavy robotu. Na obr. 2.5 je zachycen programovací systém robotu. [3]

2.2 Způsoby realizace řízení pohybu

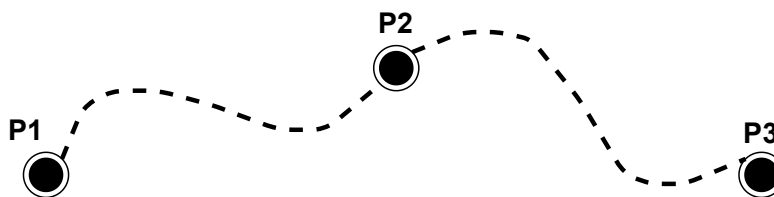
U průmyslového robotu KUKA lze řízení pohybu rozdělit na dva hlavní způsoby, tj. PTP a CP. Existuje ještě třetí způsob, tzv. MP, což je pohyb, který obsahuje hustou množinu bodů trajektorie. [4]

2.2.1 Řízení po jednotlivých bodech (PTP)

Toto řízení disponuje menšími možnostmi pohybu v prostoru, a proto se spíše hodí pro bodové robotické úlohy (vrtání, svařování, přenášení apod.). U tohoto řízení jsou definovány diskrétní body, jimiž musí robot projít, avšak pohyb mezi nimi není nijak definovaný (př. jeď z bodu X do Y přes Z). Nebo jinými slovy bodové řízení je množina bodů, kterými robot musí projít podle definované sekvence. Na obr. 2.6 je znázorněn možný postup pohybu „P1-P2-P3“ v jednotlivých bodech. Je také zřetelné, že dráha mezi jednotlivými body není definována tak, že robot při každém novém cyklu operace prochází mezi body pokaždé po jiné dráze. [4] [7] [8]



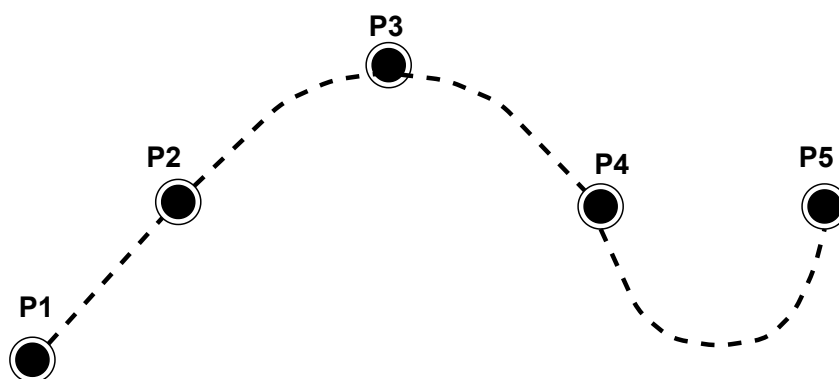
Obrázek 2.5: Ukázka programovacího/ovládacího panelu
[6]



Obrázek 2.6: Řízení pohybu metodou PTP
[4]

2.2.2 Řízení pohybu po dráze (CP)

U tohoto řízení je kromě průjezdných bodů také definováno, po jaké dráze se má těchto bodů dosáhnout (př. robot jede z bodu X do Y přes Z po kružnici). Toto řízení se realizuje pomocí „teach-in“ funkce nebo pomocí textového kódu. Řídicí počítač (interpolátor) určuje podle vyhodnocení, jaký má být počet mezihodnot na požadované dráze určité křivky. Tyto hodnoty pak zadává do polohovacího regulátoru s ohledem na nastavenou rychlost v programu. Na obr. 2.7 je znázorněn příklad pohybu CP po určité dráze. [4] [7] [8]



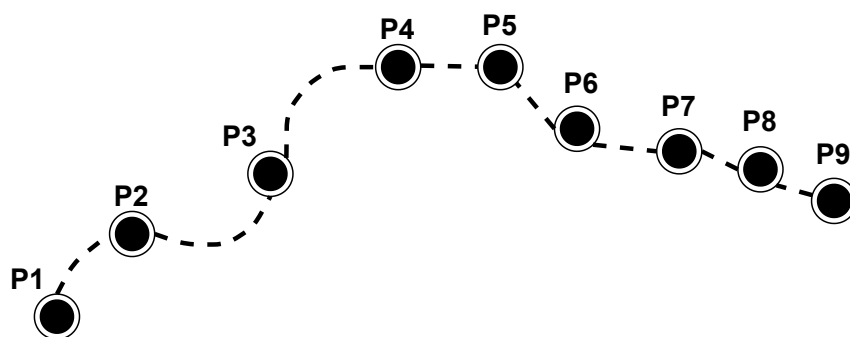
Obrázek 2.7: Řízení pohybu typu CP
[4]

2.2.3 Řízení s mnoha body (MP)

Řízení MP obsahuje pohybový program ve formě husté množiny poslopně definovaných bodů v prostoru. Tyto body se vykonávají v rychlém časovém intervalu (po 10 až 100 ms) z polohovaného regulátoru os. Tento postup znázorňuje obr. 2.8. Takovýto typ řízení pohybu se hlavně používá na stříkání barev, ale taky při broušení a dalších aplikací. [4] [7] [8]

2.3 Pohybové příkazy u průmyslového robotu KUKA

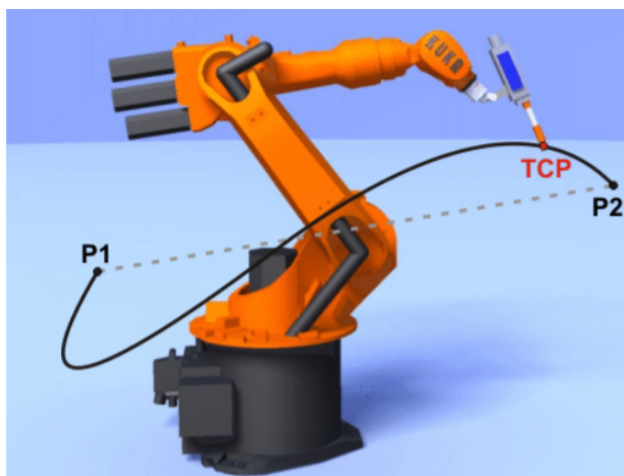
Pohybový příkaz je proces, při kterém se definuje funkce, jež podle zadaných hodnot prochází definovanými body. V technické praxi se využívají pohybové příkazy bodové, kruhové, lineární a pohyby s velkým množstvím bodů v definované dráze (tzv. PTP, CIRC, LIN, SPLINE). [4] [7] [8][9]



Obrázek 2.8: Řízení pohybu typu MP
[4]

2.3.1 Pohybový příkaz PTP

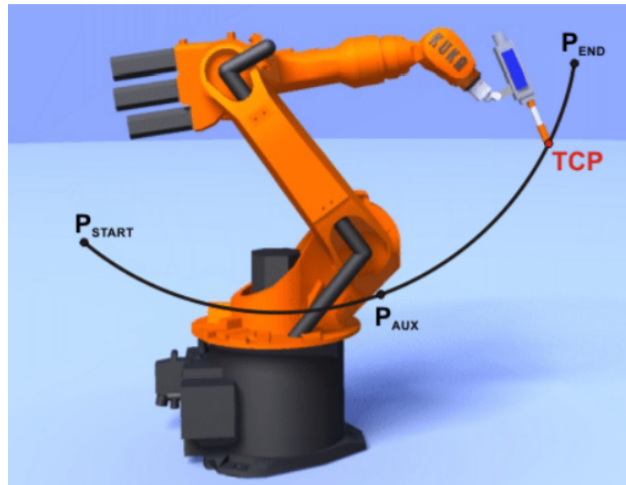
V režimu PTP se realizují přípravné pohyby robotu, kdy se vyžaduje co nejrychlejší přemístění robotu do žádané pozice při vykonávání pohybu robotu přes definované body. Tento typ pohybu umožňuje velmi plynulý pohyb bez zbytečného zpomalení. Na obr. 2.9 lze pozorovat trajektorii PTP pohybu. [4] [7] [8]



Obrázek 2.9: Pohybový příkaz PTP
[10]

2.3.2 Pohybový příkaz CIRC

Dalším typem pohybu je pohyb po kružnici. Aby řídicí systém robotu uměl tento typ pohybu realizovat, je třeba znát kromě počátečního a koncového bodu i tzv. pomocný bod, díky němuž se vytvoří pohyb po půlkružnici. Obr. 2.10 znázorňuje body počáteční a koncový (P start, P end), mezi nimiž se s označením (P aux) nachází bod pomocný. [4] [7] [8]



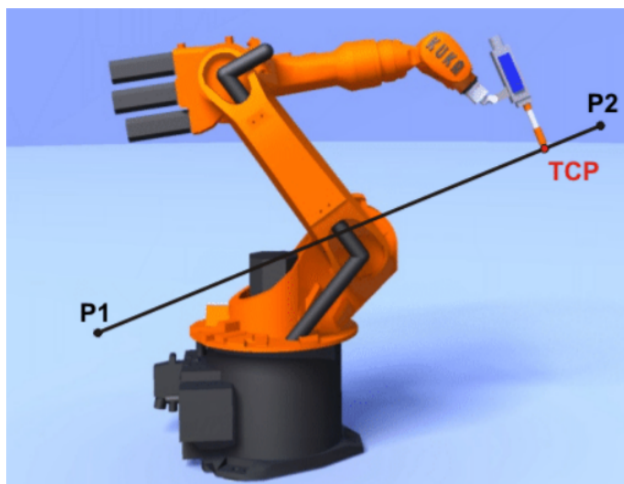
Obrázek 2.10: Pohybový příkaz CIRC
[10]

2.3.3 Pohybový příkaz LIN

Řízení robotického ramene je možné realizovat také pomocí pohybu typu LIN, což je pohyb po přímce. Při pohybu tohoto typu se rameno robotu pohybuje pomaleji, než je tomu u pohybu typu PTP. Tento typ pohybu se hlavně využívá pro operace přesného navádění koncového efektoru do žádané pozice (př. při montáži). Trajektorie pohybu LIN je znázorněná na obr. 2.11. [4] [7] [8]

2.4 Aproximace pohybových příkazů

Všeobecně problematika aproximace funkcí se zabývá způsoby nalezení méně složitých funkcí, které jsou dostatečným přiblížením funkce jiné. Na rozdíl od interpolace nemusí být u aproximace splněna podmínka, že nový odhad funkce prochází všemi definovanými funkčními hodnotami funkce aproximované. V technické praxi řízení robotických systémů se pojem aproximace funkce nazývá tzv. „blending“. Ve své podstatě jde o to, že máme v prostoru definované například tři body, u kterých chceme, aby byly spojeny po přímkách. Tomuto procesu se říká interpolace, což je proces odhadu funkce, kdy musí být splněna podmínka, že daná funkce prochází všemi definovanými body. Proces, který nastává poté, je tzv. „blending“ (aproximace), kdy chceme udělat nový odhad funkce



Obrázek 2.11: Pohybový příkaz LIN
[10]

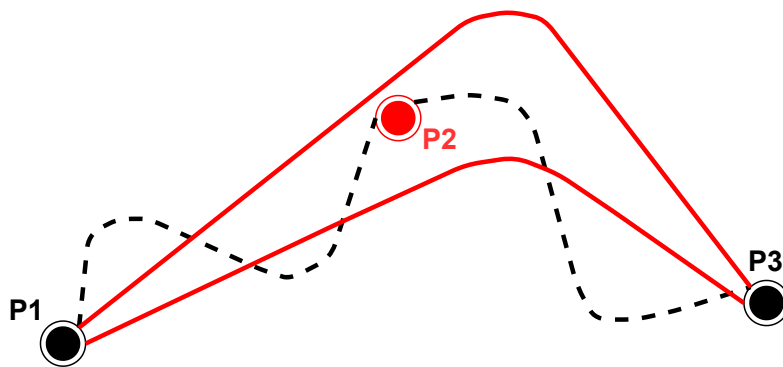
a potřebujeme, aby výsledná trajektorie byla zkrácená tak, aby robot nezastavoval v daném bodě, ale pouze okolo něj projel. Toto má za následek zrychlení pohybu robotu a úsporu energie. [4] [8] [10] [11]

2.4.1 Aproximace PTP pohybové trajektorie

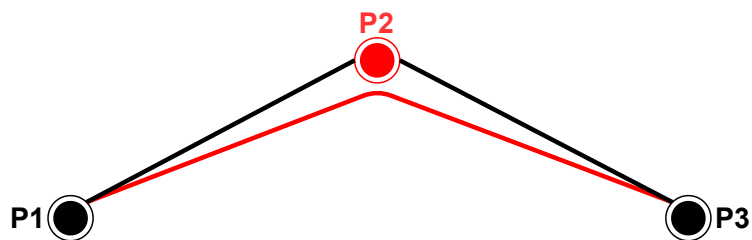
Při pohybu PTP (point-to-point) se osy robotu začínají a končí pohybovat současně. Na obrázku 2.12 je vidět pohyb PTP s aproximací červeně a bez aproximace černě. Protože pohyb typu PTP je používán k rychlým robotickým operacím např. „bin picking“. Je snaha tento proces ještě zrychlit, toho lze dosáhnout právě zjednodušením již definované funkce, což vede k vytvoření nové kratší trajektorie. Aproximace pohybu PTP umožňuje hladší a kratší průběh pohybu. V případě robotu kuka je velikost aproximace udávána v procentech (0 - 100 %). Při hodnotě 0 % dochází k tomu, že je robot naveden po nejkratší trajektorii do výchozího bodu. Pokud je zvolena jiná hodnota, výchozí trajektorie má úplně jiný charakter. [4] [7] [10] [11]

2.4.2 Aproximace LIN pohybové trajektorie

U aproximace pohybu LIN dochází k tomu, že nová funkce má velký počet společných funkčních hodnot stejných s funkcí původní. Na obr. 2.13 můžeme pozorovat, že začátek a konec trajektorie pohybu je totožný. Funkce se ale liší v průjezdném bodě „P2“, kdy u původní funkce nástrojová osa robotu zastavuje a prodlužuje se tím doba potřebná pro vykonání trajektorie. Červená čára na obr. 2.13 znázorňuje aproximovanou funkci s hladkým průběhem, kdy dochází k vytvoření rychlejší a kratší dráhy pro robotu. Při lineárním pohybu a aproximaci 100 % dochází k tomu, že robot obejde bod v polovině dráhy mezi prvním a koncovým bodem trajektorie. [4] [8] [10] [11]



Obrázek 2.12: Aproximace pohybového příkazu PTP
[10]

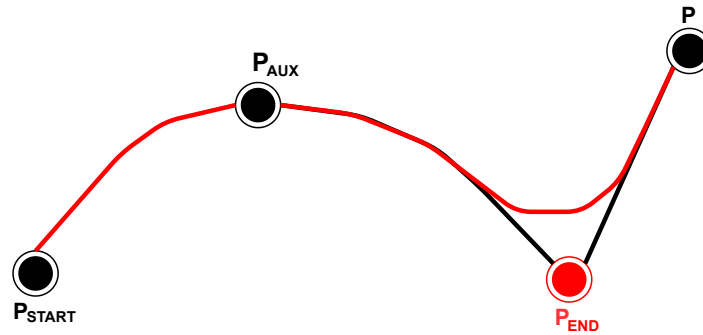


Obrázek 2.13: Aproximace pohybového příkazu LIN
[10]

2.4.3 Aproximace CIRC pohybové trajektorie

U pohybu typu CIRC, tedy pohybu po kružnici, je potřeba zadávat pomocný bod, jenž je definovaný mezi bodem startovacím a koncovým. U pomocného bodu nedochází k realizaci aproximace

viz obr. 2.14. K aproximaci dochází mezi koncovým bodem pohybu a bodem následující operace.
[4] [8] [10] [11]

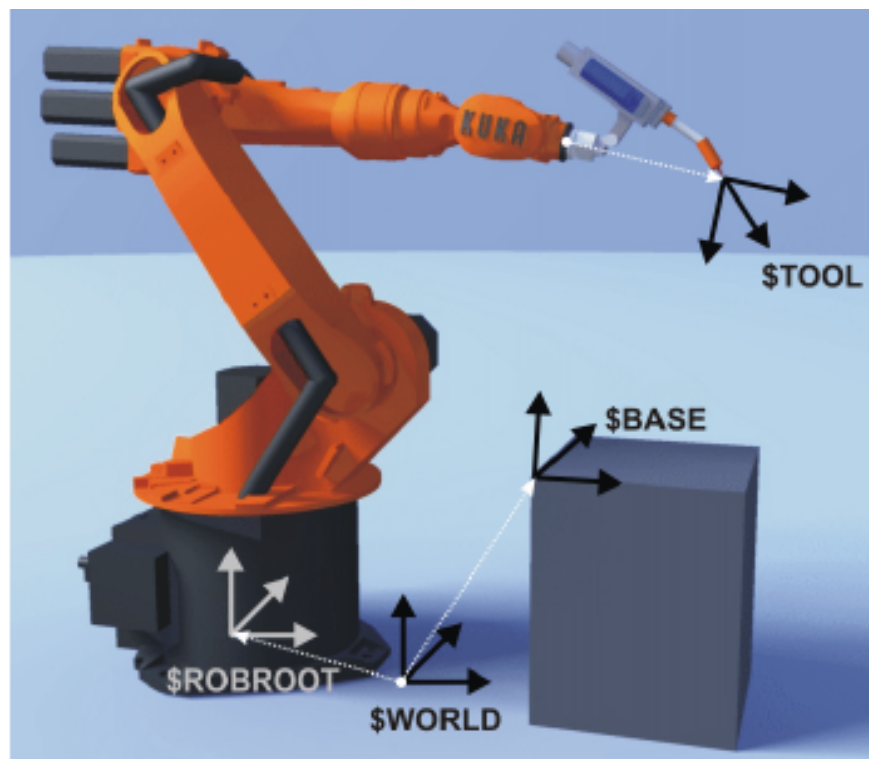


Obrázek 2.14: Aproximace pohybového příkazu CIRC
[10]

2.5 Typy souřadnicových systémů robotu

Šestiosé robotické systémy standardně pracují v kartézském souřadnicovém systému (World). Tyto systémy ale také umožňují v případě potřeby zvolit jiný souřadnicový systém, jako je například nástrojový souřadnicový systém (Tool) nebo externí souřadnicový systém (Base), který je situován mimo robotu. V případě potřeby je zde ještě možnost řídit robotu po jednotlivých osách, v takzvaném osovém souřadnicovém systému (Joint). Na obrázku. 2.15 jsou znázorněny jednotlivé souřadné systémy šestiosého robotu. [12]

- World - je definován jako světový kartézský souřadnicový systém, který je fixovaný. V defaultním nastavení je definován v patě robotu a je základem pro souřadné systémy Robroot a Base. [12] [13]
- Tool - je souřadný systém nástroje robotu umístěný ve středním bodě nástroje (TCP). Tento systém si definuje sám uživatel a je vztažen k souřadnému systému Base. [12] [13]
- Base - tento souřadný systém je vztažen ke světovému systému World a definuje polohu. [12] [13]

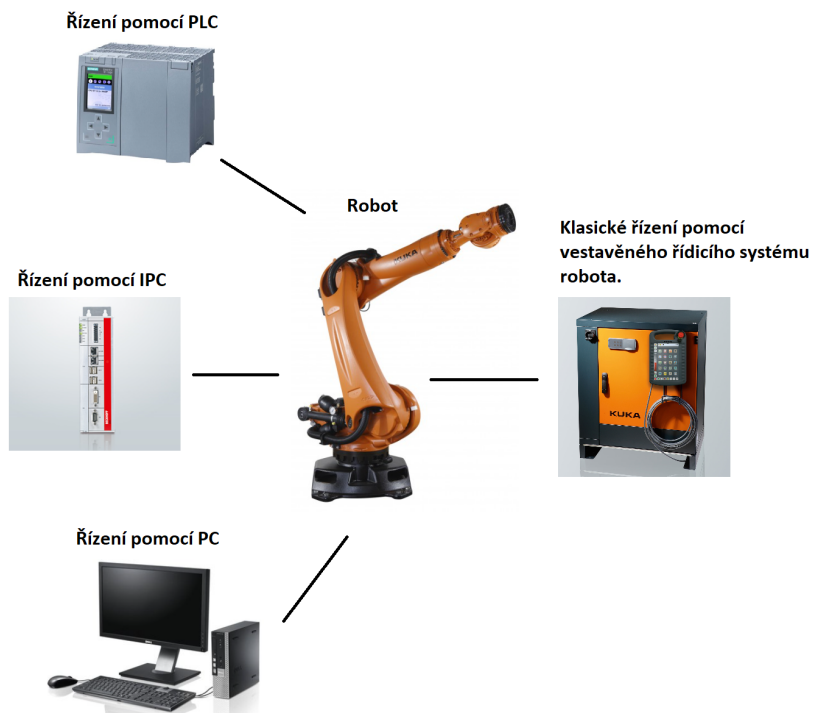


Obrázek 2.15: Souřadnicové systémy World, Tool, Base, Robroot [12]

- Robroot - takovýto souřadný systém je definován na základně robotu a je využíván jako referenční pro mechanické konstrukce robotu. A taky zároveň definuje pozici robotu vzhledem souřadnému systému World. [12] [13]

2.6 Hardwarové možnosti řízení stacionárních šestiosých průmyslových robotů

Průmyslový šestiosý robot vykonává pohyb na základě programu, který zpracovává jeho integrovaná řídicí jednotka, již dodává spolu s robotem jeho výrobce. V praxi se nejčastěji pro řízení robotu používá jeho integrovaný řídicí systém. Avšak v současné době je možnost pro speciální aplikace nahradit řídicí systém robotu klasickým PC, IPC, PLC. V případě klasického PC nebo IPC, které jsou vybaveny operačním systémem reálného času, bývají využívány pro aplikace, kdy pohyby robotu jsou vytvářeny na základě strojového vidění. Poslední možností je nahradit řídicí jednotku PLC systémem, jenž vykonává veškerý výpočetní výkon pro polohování robotu. Tato metoda se používá u aplikací, kde je potřeba sjednotit ovládání robotu s nějakým jiným strojem. Nejčastěji se jedná o robotické jednotky pro obsluhu CNC center. Možnosti řízení robotu jsou znázorněny na obr. 2.16, je zde vidět klasické řízení a další tři varianty externího řízení. [14] [15] [16]



Obrázek 2.16: Hardwarové možnosti řízení robotu

2.7 Možnosti programování stacionárních šestiosých průmyslových robotů

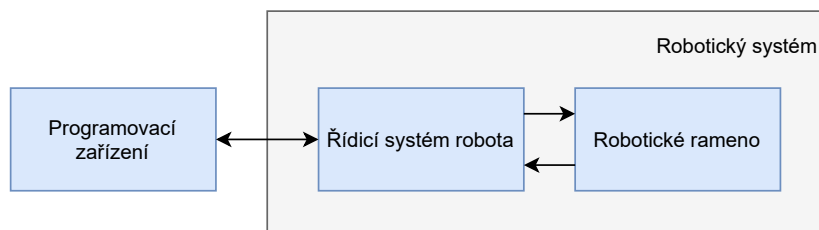
V praxi je běžně standardem, že vestavěný průmyslový robot se programuje z pendantu nebo z počítače připojeného na přímo do řídicí jednotky robotu; tato druhá možnost bývá častější z důvodů pohodlí. Obě výše zmíněné metody, u kterých se využívá k programování jazyk daného výrobce robotu, jsou přímými programovacími metodami, neboli jde o on-line programování robotu. Další možností jak naprogramovat robotu, je metoda nepřímá. Tato, jinak nazývaná off-line metoda, využívá OLP programovací jazyk a rovněž využívá simulace robotu, který je 3D znázorněním virtuálního modelu robotu. Pak existují ještě další možnosti, jak naprogramovat průmyslového robotu. Tyto metody většinou zahrnují to, že řídicí systém robotu není využíván a je nahrazen jiným hardwarem. Jednou z těchto možností je knihovna mxAutomation, která umožňuje řídit průmyslové kuka roboty pomocí PLC systémů od různých výrobců. [4] [7] [17]

2.7.1 On-line metoda programování robotu

Programování robotu online může být realizováno více způsoby. Avšak základem je vždy programovací jazyk robotu, u kuky to je KRL (Kuka Robot Language). Psaní kódu jako takového může být realizováno na pendantu nebo PC viz obr. 2.17, kde lze tyto nástroje vidět. Avšak učení jednotlivých pozic bodů v prostoru, kterých má robot dosáhnout, je realizováno buď přes pendant, PC, anebo pomocí ručního navádění robotu, kdy člověk uchopí robotu za jeho nástrojovou osu a učí ho dráhu, kterou má vykonávat. Tato metoda je využívána především u kolaborativní robotiky. Struktura postupu při on-line programování je znázorněna na obr. ??.



Obrázek 2.17: Nástroje pro on-line programování (PC,Pendant)
[4]

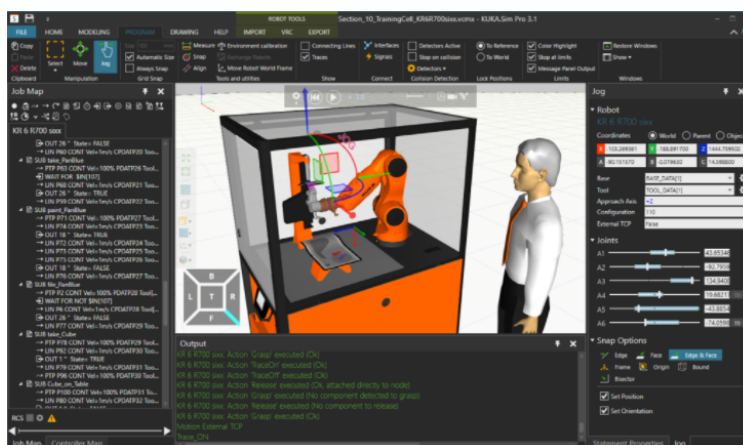


Obrázek 2.18: Znázornění postupu při on-line programování
[4]

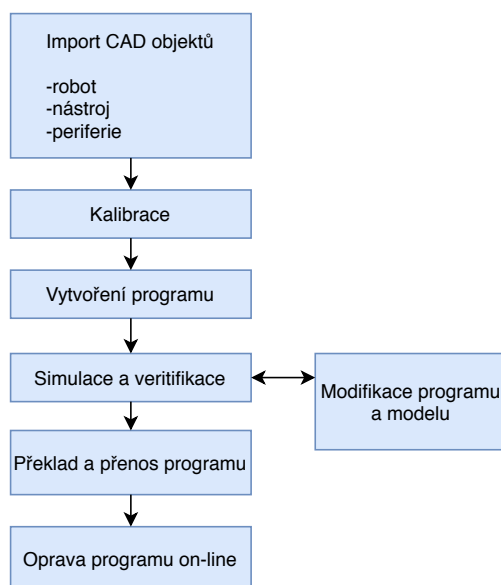
2.7.2 Off-line metoda programování robotu

Metoda off-line programování je novým způsobem, jak vytvořit a odladit program robotu. Tento způsob je založen na simulačním OLP softwaru, viz obr. 2.19. Simulace a OLP jde bok po boku. Lze mít simulaci bez OLP, ale už nejde mít OLP bez simulace. Simulace robotu je třírozměrné znázornění robotu v prostoru. Vizualně zobrazuje, jak se robot pohybuje v prostoru. Tento pohyb je vizualizací, jak robot vykonává pohyb po trajektorii z jedné souřadnice XYZ do jiné souřadnice XYZ. Výhoda off-line programování spočívá v tom, že při vytváření programu, lze odhalit spoustu kolizí mezi robotem, nástrojem upínacími přípravky a bezpečnostním ohrazením nebo jiným robotem. V simulaci lze taky analyzovat limity kloubů. Off-line programování využívá k tvorbě programu

jazyk OLP. Výsledný program je tedy převáděn do jazyka specifického pro konkrétního robota. Jinými slovy řečeno - program je vytvořen v OLP jazyce, následně je přeložen do jazyka pro daného robota (např. pro kuka robot je to KRL). Aby se OLP programování vyplatilo, musí virtuální svět odpovídat co nejvíce reálnému světu. Jinými slovy to znamená, že simulace musí být přesným modelem, který reprezentuje reálnou robotickou buňku. Pro lepší představu, jak postupovat při off-line programování, se nabízí obr. 2.20, kde je celý proces znázorněn.[17] [18]



Obrázek 2.19: Znáznornění simulačního prostředí pro off-line programování [17]



Obrázek 2.20: Znáznornění postupu při off-line programování [18]

2.7.3 Metoda programování robotu založená na mxAutomation

Nativní programovací prostředí KUKA.PLC mxAutomation zjednodušuje integraci robotů do strojních sestav. S řídicím softwarem KUKA.PLC mxAutomation lze s externím řídicím systémem (PLC) ovládat roboty na úrovni základních pohybových příkazů. To má za následek snadnou realizaci zákaznický přívětivého centralizovaného systému, který je programovatelný z jednoho druhu technologie. Další výhodou je, že jsou nadále k dispozici kvalitativně vysoké kinematické a bezpečnostně relevantní funkce řídicího systému. [19]

- Snadné programování - Při použití KUKA.PLC mxAutomation nemusí mít uživatel znalost programovacího jazyka . Stačí, když bude znát jeden z programovacích jazyků PLC podle normy IEC 61131-3. [19]
- Vysoká flexibilita - Pokud se změní požadavky ve výrobě, je možno realizovat pomocí ovládání stroje založeného na mxAutomation jakékoliv změny nebo rozšíření ovládacího softwaru. [19]
- Velké množství podporovaných dodavatelů PLC - Siemens (S7-300, S7-400, S7-1500), AllenBrodley, LabView, Beckhoff, Schneider-Electric, B&R. [19]

Kapitola 3

Analýza a popis problematiky ovládání robotu pomocí softwarových modulů

V dnešní době, kdy každý den dochází k novým a novým inovacím v průmyslové sféře, došlo k tomu, že je možné naprogramovat pohyby robotu jiným způsobem než pomocí jazyka daného výrobce robotu. A to pomocí programovacích jazyků PLC, tento způsob umožňuje to, že programátor nemusí znát programovací jazyk robotu, ale stačí mu, že zná jeden z programovacích jazyků PLC. U robotických systémů KUKA se jedná o softwarový balíček mxAutomation. Tento balíček je knihovna funkcí, která zprostředkovává komunikaci mezi PLC a řídicí jednotkou robotu a taky obsahuje funkce, které řízení robotu umožňuje. Úlohou PLC je provádět operace, včetně kinematiky a transformace pomocí mechaniky robotu. V této kapitole je popsán postup integrace a konfigurace knihovny mxAutomation do prostředí TIA portálu. Tato kapitola rovněž obsahuje popis funkcionality vybraných funkčních bloků z knihovny mxAutomation. [20] [21]

3.1 Popis systému a vytváření programu v mxAutomation a jeho zpracování

Aby program vytvořený pro PLC, například SIMATIC S7-1500 v prostředí TIA portal, komunikoval s řídicí jednotkou robotu, je potřeba, aby program vytvořený s pomocí funkcí z knihovny mxAutomation byl situován mezi bloky „KRC_ReadAxisGroup“ a „KRC_WriteAxisGroup“. Tyto dva bloky tvoří takové komunikační rozhraní mezi zařízeními. Blok read je vždy nahoře programu a druhý blok write je na konci. V ukázkovém programu, jenž společnost Siemens nabízí na svém webu, je celý program, který obsluhuje robotu, zabalen do funkce „SiemensKuka“, v němž je vloženo komunikační a programové rozhraní. Celý popis programu a zpracování dat je vidět na obr. B.1, v němž je znázorněno komunikační rozhraní, mezi kterým je program. Ve spodní části obrázku je vidět zpracování dat. Zápis dat do robotu je realizován tak, že v řídicí jednotce robotu jsou jednotlivě

uloženy příkazy a podle principu FIFO jsou přeneseny do programu paměti řídicí jednotky robotu, odkud jsou pak pohyby robotu provedeny. [20] [21]

3.2 Členění knihovny mxAutomation

Knihovna robotických funkcí pro ovládání robotu skrze PLC se skládá ze dvou částí. Knihovna bloků pro programování robotu KUKA se SIMATIC ovladačem, popřípadě jiným ovladačem pro jiné PLC. A druhou část tvoří tlumočnický blok na řídicí jednotce robotu, jenž interpretuje příkazy funkčních bloků. Knihovna mxAutomation obsahuje dva typy pohybových bloků. [20] [21]

- Bloky s předponou v názvu „KRC“- Tyto bloky jsou součástí díky vysoké kompatibilitě, ale zároveň se nechovají podle normy PLCopen. [20]
- Bloky s předponou v názvu „MC“- Bloky jsou vytvořeny podle standardu PLCopen. [20]

3.3 Integrace robotu do hardwarové konfigurace projektu TIA Portal

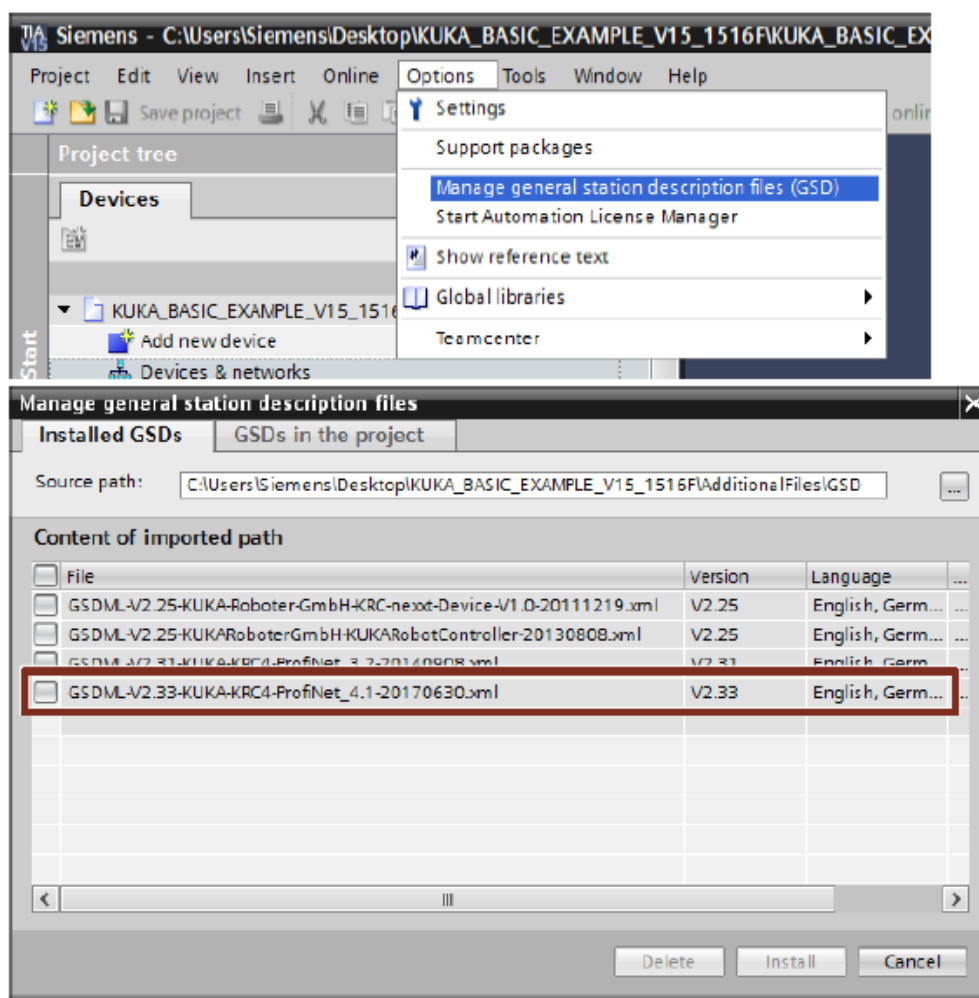
Robot KUKA je vložen jako zařízení PROFINET I/O do hardwarové konfigurace projektu TIA Portal. Robot je integrován prostřednictvím dalšího souboru GSDML, který má být vložen do projektu TIA Portal, jenž obsahuje HW popis robotu a možné datové telegramy pro výměnu dat mezi procesorem SIMATIC a robotem. [20] [21]

Instalace souboru GSDML KUKA robotu, soubor GSDML je obvykle dodáván společně s robotem, s dokumentací robotu nebo jej lze získat od služby KUKA. Soubor GSDML lze poté vložit do portálu TIA pomocí funkce správy souborů GSD, která je znázorněná na obr. 3.1. Výběr souboru GSDML závisí na verzi PROFINET, jež je nainstalovaná v řídicí jednotce robotu. Tabulka 3.1 ukazuje různé verze PROFINET s přidruženým souborem GSDML. Pro doplnění informací starší soubory GSDML je možno také použít s PROFINET 3.1 a 3.2. V tomto ohledu platí pro PROFINET 3.1 nastavení PROFINET 3.0. Pro PROFINET 3.2 platí nastavení PROFINET 3.1. [20] [21]

Integrace robotu do hardwarové konfigurace. Po integraci souboru GSDML do TIA Portal je možné robotický ovladač KUKA KR C4 v odpovídající verzi firmwaru vložit do hardwarové konfigurace v TIA Portal, pomocí funkce drag & drop. Dvojitým kliknutím myši v grafickém prostředí TIA Portal na řídicí jednotku robotu v hardwarové konfiguraci umožňuje další konfiguraci komponent pro výměnu dat s PLC SIMATIC. Aby mohla být umožněna výměna dat potřebných k řízení robotu s řadičem PLC automatu SIMATIC, musí být v hardwarové konfiguraci s řídicím systémem robotu nastaven telegram potřebný k vzájemné výměně dat. Pro tento účel musí být telegram pro 2032 digitálních vstupů a výstupů integrován ze skupiny telegramu do pod komponenty řídicí jednotky robotu v hardwarové konfiguraci pomocí funkce drag & drop. [20] [21]

Pomocí vhodného PLC SIMATIC je možné za pomoci knihovny bloků KUKA.PLC mxAutomation najednou ovládat až pět průmyslových robotů KUKA. Vymezení možností je definováno v

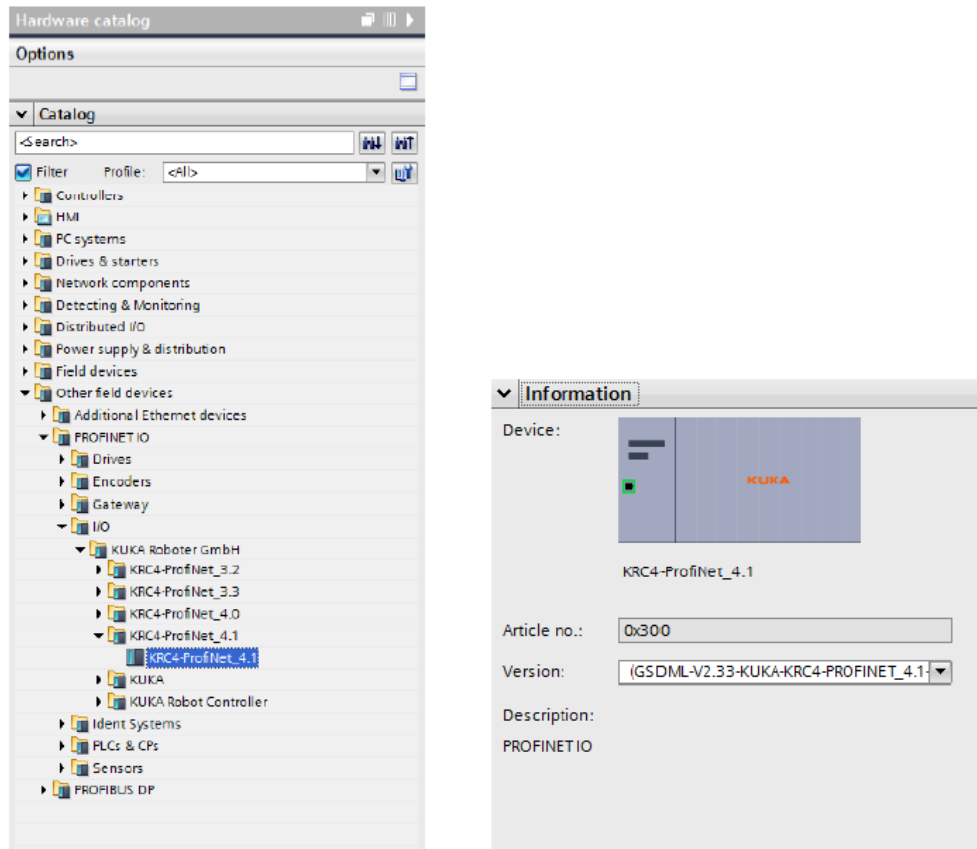
knihovně robotických funkcí. Retenční paměť je primárním separačním faktorem v procesoru PLC SIMATIC. Čím větší je množství pozic robotů, které je třeba uložit, o to více je zapotřebí větší velikost retenční paměti. Velikost paměti remanence však může být rozšířena o odpovídající rozšiřující modul, což by pomohlo ještě menším řadičům ukládat více pozic robotických ramen a teoreticky ovládat více než pět robotických systémů. V datovém bloku dodávaném s knihovnou robotických bloků KUKA.PLC mxAutomation je možno použít pět datových polí pro výměnu dat mezi PLC SIMATIC a robotem. Data jsou přiřazena danému robotu pomocí definovaných telegramových adres v místě IO procesoru SIMATIC, jak jsou definovány v hardwarové konfiguraci. Funkční bloky a funkce robotů knihovny mxAutomation jsou přiřazeny jednotlivým robotickým ramenům v uživatelském programu rozhraní blokového vstupu AxisGroupIdx, který prezentuje index datového pole v datovém bloku MxADBRobots viz obr. 3.3. [20] [21]



Obrázek 3.1: Výběr a instalace souboru GSDML „KUKA-KRC4-ProfiNet“ [20]

Tabulka 3.1: Různé verze PROFINETU s přidruženým souborem GSDML [20]

Verze PROFINET	soubor GSDML	Název zařízení (v souboru GSDML)
1.0	GSDML-V2.25-KUKA Robot GmbH-KR C4-Device- V1.0-20121030.xml	KUKA KRC nexxt Zařízení V8.1
2.0, 2.1, 2.2	GSDML-V2.25-KUKA Robot GmbH-KR C4-Device- V1.0-20121030.xml	KUKA KRC nexxt Zařízení V8.1 KUKA KRC nexxt Zařízení V8.2
2.3	GSDML-V2.31-KUKA-KRC4- PROFINET_2.3-20140704.xml	KRC4- PROFINET_2,3
3.0	GSDML-V2.25-KUKA Robot GmbH-KR C4-Device- V1.0-20121030.xml	KUKA KRC nexxt Zařízení V8.1 KUKA KRC nexxt Zařízení V8.2 Zařízení KR C4 V8.3
3.1	GSDML-V2.25-KUKARoboterGmbH- KUKARobotController- 20130808.xml	Kr C4 zařízení V8.3
3.2	GSDML-V2.31-KUKA-KRC4- ProfiNet_3.2-20140908.xml	KRC4-ProfiNet_3,2
3.3	GSDML-V2.32-KUKA-KRC4- ProfiNet_3.3-20161027.xml	KRC4-ProfiNet_3.3
4.0	GSDML-V2.32-KUKA-KRC4- ProfiNet_4.0-20170412.xml	KRC4-ProfiNet_4.0
4.1	GSDML-V2.33-KUKA-KRC4- ProfiNet_4.1-20170612.xml	KRC4-ProfiNet_4.1



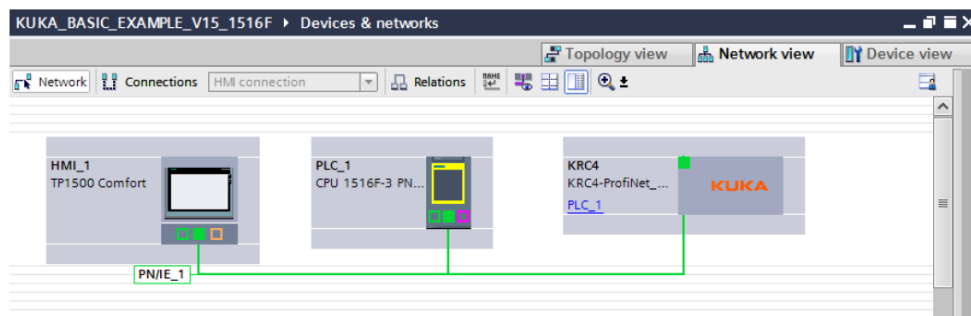
Obrázek 3.2: Přidání řídicí jednotky robotu do hardware konfigurace [20]

MxADBRobots				
	Name	Data type	Offset	Start value
1	Static			
2	KRC_AxisGroupRefArr	Array[1..5] of *AXIS_GROUP_REF*	0.0	
3	KRC_AxisGroupRefArr[1]	*AXIS_GROUP_REF*	0.0	
4	KRC_AxisGroupRefArr[2]	*AXIS_GROUP_REF*	736.0	
5	KRC_AxisGroupRefArr[3]	*AXIS_GROUP_REF*	1472.0	
6	KRC_AxisGroupRefArr[4]	*AXIS_GROUP_REF*	2208.0	
7	KRC_AxisGroupRefArr[5]	*AXIS_GROUP_REF*	2944.0	
8	PLC_SYS_TICK_CNT	Dint	3680.0	0
9	PLC_TICKS_PER_SEC	Int	3684.0	0

Obrázek 3.3: Datový blok pro pět robotů v datovém bloku MaxDB Robots [20]

3.3.1 Propojení robotu a PLC SIMATIC

Propojení robotu s PLC je realizováno tak, že musí být řídicí jednotka robotu a PLC SIMATIC vzájemně propojena prostřednictvím připojení PROFINET viz obr. 3.4. Za tímto účelem musí mít jednotlivé systémy v hardwarové konfiguraci přiřazeny vhodné IP adresy. Je taky možnost libovolně do konfigurace integrovat zařízení HMI, kterým lze ovládat a monitorovat funkce robotu. [20] [21]



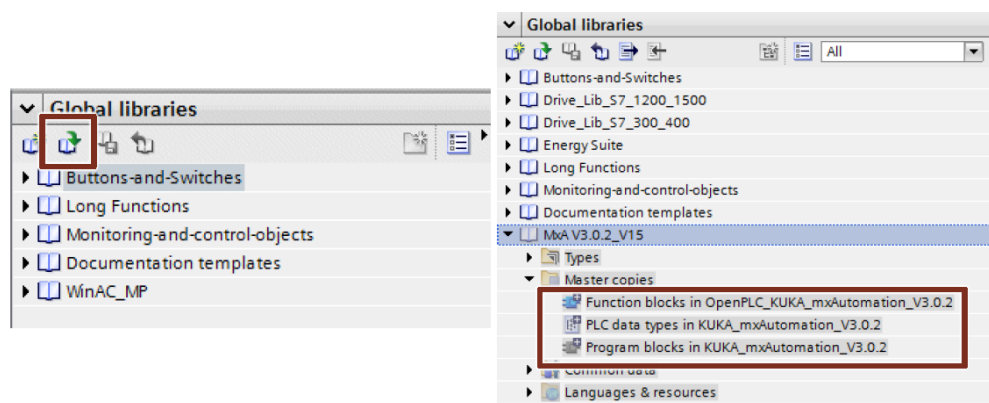
Obrázek 3.4: Vytvoření komunikace mezi robotem a PLC

[20]

3.4 Import knihovny robotických bloků mxAutomation a použití v programu

Vložení knihovny bloků KUKA.PLC mxAutomation je důležité proto, aby bylo možno vytvořit uživatelský program pro ovládání robotu. U knihovny bloků KUKA.PLC mxAutomation je třeba funkční bloky knihovny přenést do projektu TIA Portal. Knihovna bloků KUKA.PLC mxAutomation pro TIA Portal je dodávána jako globální knihovna funkcí, ze které je možné pomocí funkce drag & drop vkládat všechny funkční bloky knihovny do rozhraní projektu v TIA Portal. Aby byla globální knihovna použitelná pro TIA Portal, je třeba rozbalit archiv globální knihovny na vhodné místo na pevném disku PC, kde bude tvorba projektu robotické aplikace realizována. Po importování je možné otevřít globální knihovnu prostřednictvím přístupových funkcí TIA Portal, viz obr. 3.5, kde je knihovna bloků KUKA.PLC mxAutomation uložena jako šablona v globální knihovně. Přenos bloků KUKA.PLC mxAutomation do uživatelského programu je možné přenést snadným přetažením z knihovny funkčních bloků. Potřebné datové typy PLC SIMATIC i funkční bloky musí být přeneseny z kopírovacích šablon globální knihovny do projektu TIA Portal. Všechny šablony jsou ve složce „Hlavní kopie“. Jako první je potřeba přetáhnout objekt „Uživatelské datové typy“ do složky „Datové typy PLC“ v projektu TIA Portal. Zde se automaticky vytvoří podsložka, která obsahuje všechny datové typy PLC knihovny bloků KUKA.PLC mxAutomation. Poté je možno zbývající objekty do složky „Programové bloky“ projektu TIA Portal pro danou robotickou apli-

kaci. Zde se automaticky vytvoří podsložka, která obsahuje všechny funkční bloky knihovny bloků KUKA.PLC mxAutomation. [20] [21]



Obrázek 3.5: Globální knihovna funkčních bloků
[20]

3.5 Základní členění programu

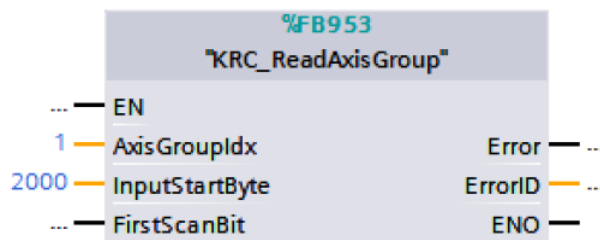
Celý program pro robotickou aplikaci je integrován do funkce (FC) pro lepší přehled a strukturu programu s názvem „SiemensKuka“. V této funkci jsou potřebné bloky pro robotu z knihovny KUKA.PLC mxAutomation a jsou volány další funkční bloky, které obsahují speciální pohybové příkazy pro robotické rameno. Funkce „SiemensKuka“ má následující hierarchii. Čtení hodnot z robotického ramene za pomoci funkčního bloku „KRC_ReadAxisGroup“. Vykonávání elementárních funkcí robotu je zprostředkováno pomocí funkčního bloku „KukaControl“. Tento funkční blok musí být proveden, aby mohlo dojít k provedení jednoduchého pohybu typu pick & place. Zápis informací do řídicí jednotky robotu je realizován pomocí funkčního bloku „KRC_WriteAxisGroup“. Je také důležité zmínit, že funkce „SiemensKuka“ kromě toho, že umožňuje lepší strukturování, měla by být brána v úvahu jako call-FC. Jinými slovy to znamená, že obsahuje přístupy do globálních databází, a proto nemůže být použita jako prvek knihovny. [20] [21]

3.6 Typy funkcí a funkčních bloků pro ovládaní robotu

V této kapitole budou podrobně rozebrány nejdůležitější funkce a funkční bloky, které knihovna KUKA.PLC mxAutomation obsahuje. Součástí popisu bude všeobecný popis jednotlivých bloků, respektive jejich funkcionality, a taky budou tyto bloky popsány z hlediska nastavovacích procedur. [20] [21]

3.7 Funkční blok KRC_ReadAxisGroup

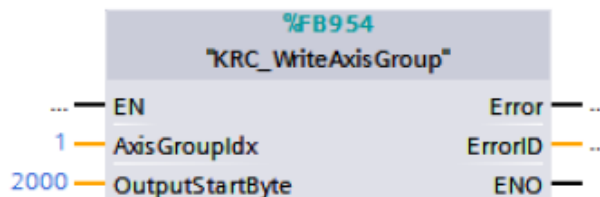
Funkční blok „KRC_ReadAxisGroup“ je možné vidět na obr. 3.6, má označení „FB953“ z knihovny KUKA.PLC mxAutomation a má za úkol čtení informací z řídicí jednotky robotického ramene do vnitřního datového úložiště knihovny bloků KUKA.PLC mxAutomation v procesorové jednotce SIMATIC. Díky tomu je umožněno poskytování dat z řídicí jednotky robotu do uživatelského programu, který je nahrán v daném PLC. Tato data mohou být dále poskytnuta ostatním funkčním blokům z knihovny KUKA.PLC mxAutomation, které jsou umístěny v uživatelském programu. Realizace přiřazení dat k danému robotickému systému, jenž je připojen k PLC SIMATIC, je realizováno za pomoci vstupu „AxisGroupIdx“, který je roven indexu datového pole v datovém bloku (DB) „MxADBRobots“ a vstupu „InputStartByte“, ve němž je umístěna adresa telegramu robotu a je odeslán z vstupně výstupní oblasti procesoru PLC do daného funkčního bloku. [20] [21]



Obrázek 3.6: Funkční blok pro čtení dat z robotu
[20]

3.8 Funkční blok KRC_WriteAxisGroup

Jakmile PLC zpracuje program robotu, blok „KRC_WriteAxisGroup“ (FB954) z knihovny KUKA.PLC mxAutomation přenese data z datového bloku „MxADBRobots“ do robotu definovaného adresou telegramu. Tento blok je vidět na obr. 3.7, kde jsou znázorněny jeho vstupní a výstupní proměnné. [20] [21]

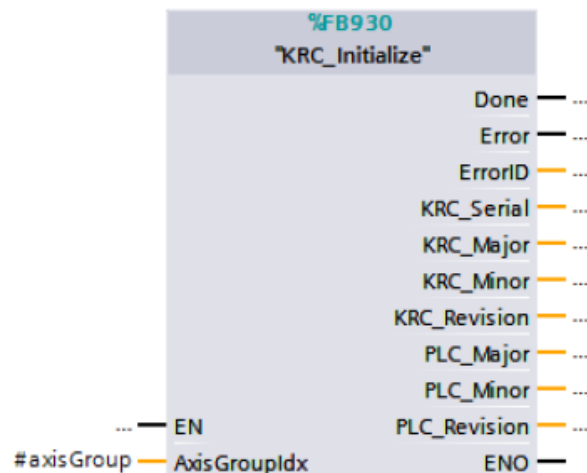


Obrázek 3.7: Funkční blok pro zápis dat do robotu
[20]

3.9 Funkční blok KRC_Initialize

Před výměnou dat mezi PLC SIMATIC a robotickým systémem kuka musí být inicializováno rozhraní KUKA.PLC mxAutomation. Z tohoto důvodu je potřeba zavolat blok „KRC_Initialize“ (FB930), který je vidět na obr. 3.8, kde je možné sledovat množství výstupních proměnných, jež jsou popsány v této kapitole. Pokud je inicializace rozhraní úspěšně provedena, tak dojde k tomu, že je zasláno hlášení zpět do uživatelského programu pomocí výstupu „Hotovo“. [20] [21] Kromě toho blok vrací následující informace:

1. Sériové číslo řídicí jednotky robotu na výstupu „KRC_Serial“
2. Verze tlumočníka na výstupech:
 - „KRC_Major“
 - „KRC_Minor“
 - „KRC_Revision“
3. Verze knihovny bloků na výstupech:
 - „PLC_Major“
 - „PLC_Minor“
 - „PLC_Revision“



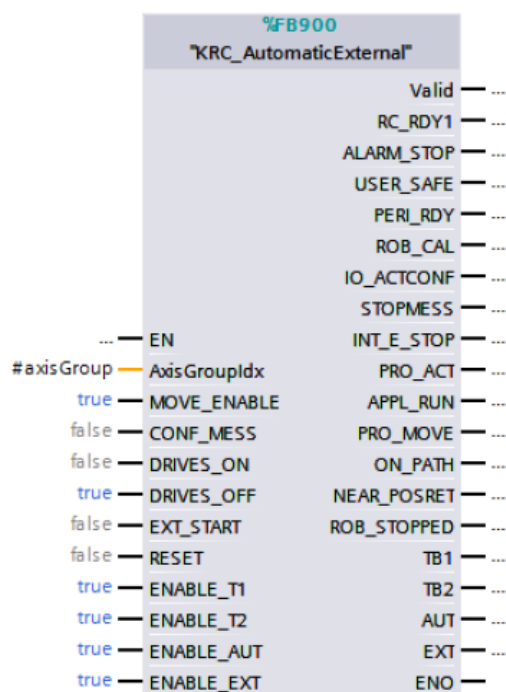
Obrázek 3.8: Funkční blok pro inicializaci rozhraní KUKA.PLC mxAutomation
[20]

Tabulka 3.2: Popis důležitých vstupních hodnot bloku KRC_AutomaticExternal [20]

Vstup	Popis
AxisGroupIdx	Index skupiny os
MOVE_ENABLE	TRUE = uvolnění pro robotu
DRIVES_OFF	TRUE = deaktivace pohonů robotu
ENABLE_T1	TRUE = uvolnění pro režim T1 MOVE_ENABLE
ENABLE_T2	TRUE = uvolnění pro režim T2 MOVE_ENABLE
ENABLE_AUT	TRUE = uvolnění pro režim AUT MOVE_ENABLE
ENABLE_EXT	TRUE = uvolnění pro režim EXT MOVE_ENABLE

3.10 Funkční blok KRC_AutomaticExternal

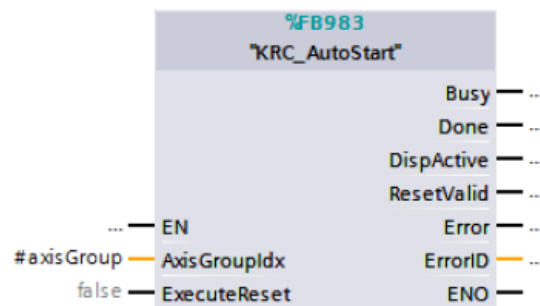
Pro navázání komunikace mezi PLC systémem SIMATIC a řídicí jednotkou robotické stanice musí být rozhraní „Automatic External“ aktivováno pomocí funkčního bloku „KRC_AutomaticExternal“ (FB900), který je vidět na obr. 3.9, a také musí být načteny signály rozhraní. Tento funkční blok může být vyvolán pouze jednou na skupinu os. Jednotlivé prvky funkčního bloku jsou popsány v tabulce 3.2. [20] [21]



Obrázek 3.9: Funkční blok pro navázání spojení mezi PLC a robotem kuka
[20]

3.11 Funkční blok KRC_AutoStart

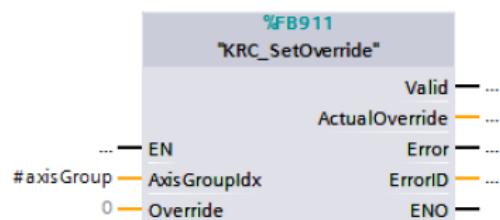
Funkční blok „KRC_AutoStart“ viz obr. 3.10 uspořádá signály rozhraní pro funkční blok „KRC_AutomaticExternal“ ve správné sekvenci pro spuštění robotu. Pokud funkční blok ohlásí chybu, která se projeví tím, že robot se nezapne a aktivuje se výstup „Chyba“. Chybový kód aktuální chybové zprávy je možné přečíst na výstupu „ErrorID“. [20] [21]



Obrázek 3.10: Funkční blok pro uspořádání signálu
[20]

3.12 Funkční blok KRC_SetOverride

Funkční blok „KRC_SetOverride“ viz obr. 3.11 (FB911) říká, jaké má být procentuální přepsání naprogramované rychlosti robotu. U příkazů pohybu souvisejících s osou přepis působí na zadanou rychlost osy. U příkazů pohybu souvisejících s cestou přepis působí na zadanou rychlost dráhy. [20] [21]

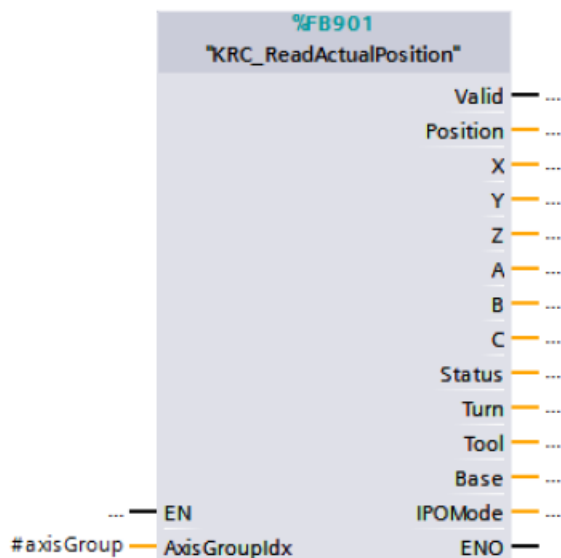


Obrázek 3.11: Funkční blok pro nastavení rychlosti robotu
[20]

3.13 Funkční blok KRC_ReadActualPosition

Funkční blok „KRC_ReadActualPosition“, který je vidět na obr. 3.12, přečte kartézskou polohu nástrojové osy robotu vzhledem k stávajícímu vybranému souřadnicovému systému, který si zvolí

uživatel. Zvolená aktuální poloha v definované struktuře „E6POS“ je odeslána přes výstupní proměnnou „Position“. Tato proměnná má v sobě obsažené všechny jednotlivé pozice v prostoru a rotaci kolem souřadnicových kloubů, jež jsou vydávány na výstupech X až Z a A až C stejně jako výstupní proměnné Status a Turn. Výstupy Tool a Base se používají k výstupu aktuálně aktivního nástroje a čísla aktuálně vybraného souřadnicového systému, který vybral uživatel. [20] [21]



Obrázek 3.12: Funkční blok pro čtení kartézských souřadnic nástrojové osy robotu

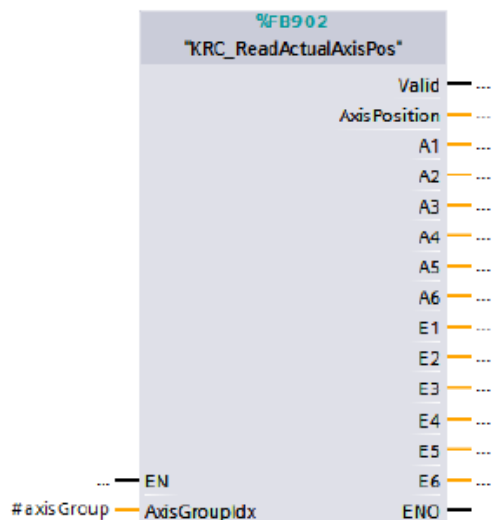
[20]

3.14 Funkční blok KRC_ReadActualAxisPos

Funkční blok „KRC_ReadActualAxisPos“, který je vidět na obr. 3.13, slouží k tomu, aby četl aktuální polohy os robotického systému. Pozice kloubů nebo respektive os robotu v definované struktuře „E6AXIS“ se odesílá na výstupu „AxisPosition“. Tato proměnná obsahuje všechny jednotlivé hodnoty, které čekají na výstupech A1 až A6 a E1 až E6. Výstupy E1 až E6 představují aktuální pozici přídatných os, která může být také ovlivněna ovladačem robotu. [20] [21]

3.15 Funkční blok KRC_Jog

Robotem je možné pohybovat v režimu krokování za pomoci funkčního bloku „KRC_Jog“, který je na obr. 3.14. Vstupní proměnnou „MoveType“ lze zvolit, pokud se má robotickými osami procházet přímo, axiálně, anebo pokud se má v kartézském souřadnicovém systému robotu pohybovat středem nástroje robotu (TCP), to znamená špičkou nástroje. Jestliže je funkce vyvolána během pohybu

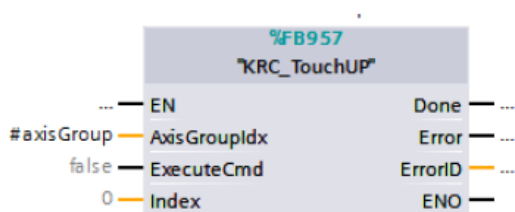


Obrázek 3.13: Funkční blok pro čtení pozic natočení jednotlivých os robotu [20]

robotu, jsou veškeré aktivní a vyrovňovací příkazy zrušeny, robotická stanice je zpomalena a následně je pohyb vykonán. Tato funkce může současně pohybovat několika osami robotu najednou. [20] [21]

3.16 Funkční blok KRC_TouchUP

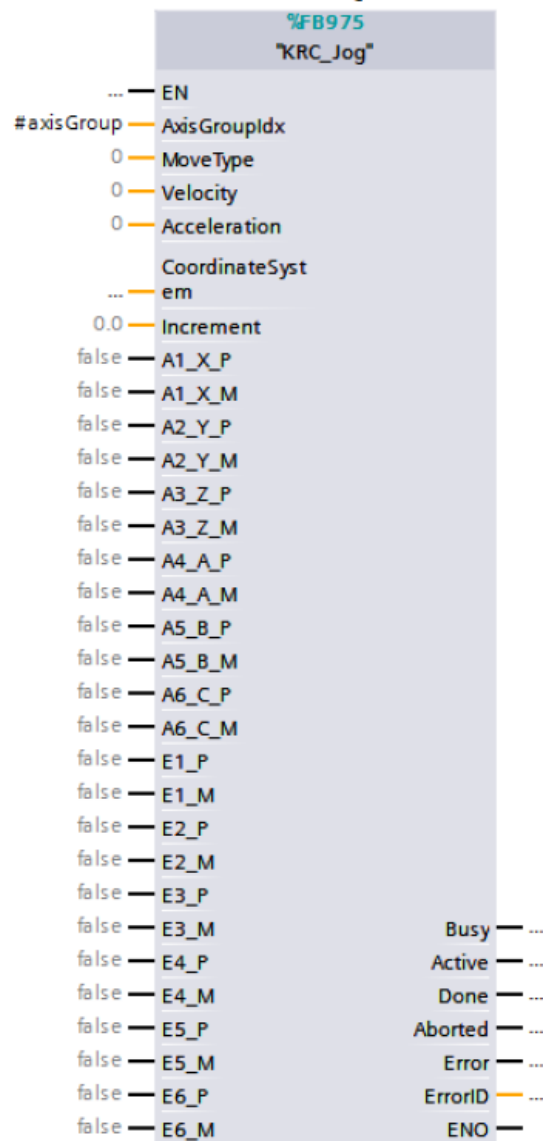
Funkční blok „KRC_TouchUP“, jenž je znázorněn na obr. 3.15, slouží k tomu, aby se naučil (zapamatoval si pozici robotu) aktuální pozici robotu v prostoru. Místo pozice a označení souřadného systému, ke kterému se daná pozice vztahuje, se ukládá do datového bloku „MxADBPosition“. Místo uložení dat v poli pozic datového bloku je definováno v „Index“. [20] [21]



Obrázek 3.15: Znázornění funkčního bloku KRC TouchUP [20]

3.17 Funkční blok KRC_Interrupt

Všechny příkazy, které řídicí jednotka robotu vykonává, jsou přerušeny funkčním blokem „KRC_Interrupt“. Díky tomu se v řídicí jednotce robotu vykoná instrukce BRAKE nebo BRAKE-F. Jestliže je



Obrázek 3.14: Znázornění funkčního bloku KRC Jog
[20]

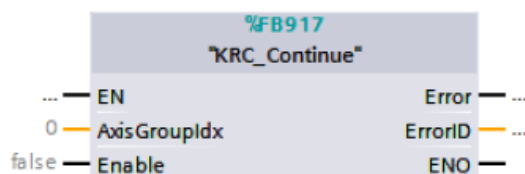
aktivní instrukce BRAKE nebo BRAKE-F, nejsou prostřednictvím rozhraní mxAutomation zpracovávány ani vykonávány žádné další instrukce. Program nebude vykonáván, dokud nebude zavolán blok „KRC_Continue“. V tomto případě může být program zrušen pouze nastavením proměnné „RESET“ na funkčním bloku „KRC_AutomaticExternal“. [20] [21]



Obrázek 3.16: Znázornění funkčního bloku pro přerušení vykonávání programu [20]

3.18 Funkční blok KRC_Continue

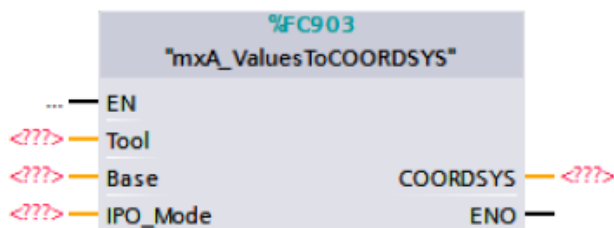
Tento funkční blok „KRC_Continue“, který se nachází na obr. 3.17, slouží k tomu, aby po jeho zavolání bylo možno pokračovat v programu po přerušení způsobeném blokem „KRC_Interrupt“. [20] [21]



Obrázek 3.17: Znázornění funkčního bloku pro pokračování programu po přerušení [20]

3.19 Funkční blok mxA_ValuesToCOORDSYS

Jednotlivé prvky se zapisují do struktury COORDSYS pomocí funkce „mxA_ValuesToCOORDSYS“ viz obr. 3.18. Obsahuje informace o tom, které souřadnicové systémy by měly být vybrány. [20] [21]



Obrázek 3.18: Znázornění funkčního bloku, který umožňuje výběr souřadného systému robotu [20]

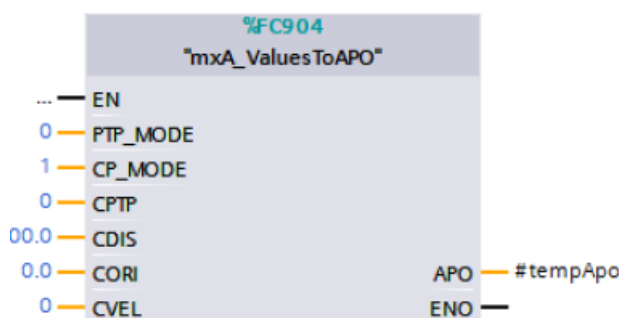
Tabulka 3.3: Popis důležitých vstupních hodnot bloku mxA_ValuesToCOORDSYS [20]

Vstup	Popis
Tool	Výběr čísla souřadného systému nástroje nebo aktivního nástroje.
Base	Výběr čísla základního souřadnicového systému, který má být použit.
IPO_Mode	Volba režimu interpolace, tj. Zda je nástroj namontován na montážní přírubě nebo zda stojí pevně.

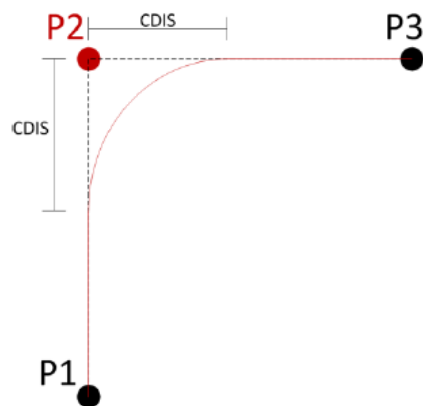
3.20 Funkční blok mx_AValuesToAPO

Funkce „mxA_ValuesToAPO“, která je zobrazena na obr. 3.19, slouží k tomu, aby uživatel mohl definovat typ pohybu, který má robot vykonat. Uživatel tímto blokem říká, jakým typem interpolace nebo aproximace bude vytvořena dráha robotu. Tento blok obsahuje parametry „blendingu“ pro pohybové příkazy. V tabulce 3.4, jsou popsány jednotlivé funkcionality vstupních proměnných bloku. [20] [21]

Vysvětlení funkce tohoto bloku je následovné. Pokud se stane, že máme dva po sobě následující lineární pohyby respektive pohyby po přímkách, tak by měl být proveden tzv. „blendig“, což je nový aproximovaný odhad funkce. Abychom toho dosáhli, musí být funkční blok nastaven tak, jak je znázorněno na obr. 3.19. Proměnná s názvem „APO“ nyní obsahuje informace o tom, že by měl být prováděn lineárními a kruhovými pohyby ($CP_MODE = 1$). Na vstupní proměnné s názvem CDIS je třeba zadat požadovanou vzdálenost (zde: 100 mm). To má za následek následující trajektorii, kterou ukazuje obr. 3.20. [20] [21]



Obrázek 3.19: Znázornění funkčního bloku, který definuje typ pohybové funkce [20]



Obrázek 3.20: Znázornění trajektorie robotu
[20]

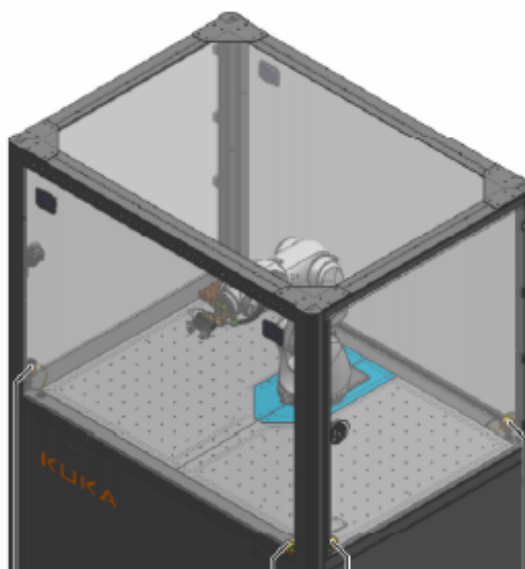
Tabulka 3.4: Popis důležitých vstupních hodnot bloku mxA_ValuesToAPO [20]

Výstup	Popis
PTP_MODE	<p>Určuje, zda a jak je vyblendován, cílový bod pohybu PTP.</p> <ul style="list-style-type: none"> • 0 Bez „blendingu“ (výchozí) • 1: Způsobí „blending“ cílového bodu. <p>Pro „blending“, PTP-PTP je dostatečná specifikace CPTP.</p> <p>Při „blending“ PTP-CP, tj. Když příkaz LIN nebo CIRC nastavený, jako „blending“ pro funkci PTP, je třeba zadat další parametr „blendingu“:</p> <ul style="list-style-type: none"> • 2: Směšování PTP-CP s parametry vzdálenosti • 3: Míchání PTP-CP s parametry orientace • 4: Míchání PTP-CP s parametry rychlosti
CP_MODE	<p>Určuje, zda a jak je cílový bod pohybu CP(LIN, CIRC) blendován.</p> <ul style="list-style-type: none"> • 0 Bez směsi (výchozí) • 1: „blending“ parametry vzdálenosti • 2: „blending“ parametry orientace • 3: „blending“ parametry rychlosti
CPTP	<p>Vzdálenost prolnutí jako procento pro pohyb z bodu na bod, od kterého by měl začít „blending“ nejdříve, před cílovým bodem. Maximální vzdálenost 100% označuje poloviční vzdálenost mezi počátečním a koncovým bodem.</p>
CDIS	<p>Vzdálenost k cílovému bodu v milimetrech, první bod, od kterého by měl začít „blending“.</p>
CORI	<p>Parametr Orientace ve stupních, první bod, od kterého může začít „blending“, když dominantní úhel orientace klesne pod zadanou hodnotu do cílového bodu.</p>
CORI	<p>Určení rychlosti v procentech, první bod, od kterého může začít „blending“ ve fázi zpomalení na cílový bod.</p>

Kapitola 4

Definování a popis vybrané laboratorní úlohy

Tato kapitola obsahuje popis robotického pracoviště ready2_educate, které je vybaveno KUKA robotem. Pracoviště je vybaveno různými komponenty, které budou popsány v následujících podkapitolách. Samotné pracoviště je vyobrazeno na obr. 4.1, kde je vidět buňka pouze s robotem. Součástí robotické buňky jsou i přípravky, pomocí kterých lze demonstrovat různé úlohy. Poslední kapitola se věnuje definování úlohy, již má robot vykonávat. Při realizaci této úlohy budou použity právě tyto přípravky, které jsou součástí výukového robotického stanoviště ready2_educate. [22]



Obrázek 4.1: Robotické pracoviště KUKA ready2_educate
[22]

4.1 Jednotlivé komponenty robotického stanoviště

Robotické stanoviště KUKA ready2_educate je složeno z mnoha komponent. Tou nejdůležitější částí je robotické rameno KR 3 R540, které je vybaveno řídicím systémem KR C4 compact. robotu lze nastavovat pomocí ovládací konzole, což je jinými slovy smartPAD. Na nástrojové ose má robot připevněno pneumatické chapadlo. Všechny tyto zmíněné komponenty a další budou zpracovány v následujících podkapitolách. [22]

4.1.1 Řídicí systém robotu KR C4 compact

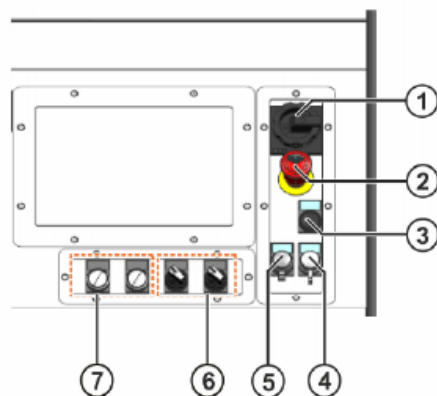
Řídicí jednotka KR C4, viz obr. 4.2, je umístěna ve spodní části robotické buňky, kde se nachází úložný prostor a kde je mimo jiné kromě řídicí jednotky uložen také rozvaděč s PLC. Dostat se do úložného prostoru k řídicí jednotce je možné odstraněním zadní stěny nebo stěn bočních. Ke konektorovému rozhraní řídicí jednotky se dostaneme po odstranění zadní stěny. Pod rozvaděčem se nachází řídicí jednotka a její konektorová část. Řídicí jednotka KR C4 compact nabízí integraci robotického řízení, PLC řízení, CNC řízení (pro obráběcí aplikace), bezpečnostní řízení. Všechny tyto systémy řízení mají společnou datovou základnu a infrastrukturu. Systém umožňuje PLC celkový přístup k I/O řídicímu systému. Proměnné jako je pozice os nebo rychlosti lze načítat a zpracovávat z funkčních modulů. [22]



Obrázek 4.2: Řídicí jednotka KR C4 compact
[22]

4.1.2 Ovládací konzole

Ovládací konzole se skládá ze dvou částí, a to z KUKA smartPAD, který je popsán v kapitole 3.1.3, a ovládacích a zobrazovacích prvků, jež ukázány na obrázku 4.3. Mezi tyto prvky patří například tlačítko „Nouzové zastavení“ a taky tlačítko „Quit“. Všechny ovládací prvky jsou popsány v tabulce 4.1, kde je uvedeno číslo, které reprezentuje daný prvek, a vedle něho je stručný popis. Ovládací prvky jsou umístěny ve spodní části robotické buňky z její přední strany. Kromě ovládacích prvků je zde ještě zobrazovací HMI panel, jemuž se věnuje kapitola 3.1.4. [22]



Obrázek 4.3: Ovládací a zobrazovací prvky
[22]

Tabulka 4.1: Popis k jednotlivým ovládacím prvků robotické stanice [22]

Číslo popisku	Popis
1.	Hlavní vypínač - hlavní vypínač slouží k vypnutí a zapnutí celé robotické buňky.
2.	Tlačítko NOUZOVÉHO ZASTAVENÍ s klíčem - pomocí přístroje NOUZOVÉHO ZASTAVENÍ se zastaví všechny pohyby.
3.	Tlačítko 'Quit' pro potvrzení ochrany obsluhy
4.	Přípojka USB
5.	Přípojka KSI
6.	Páčkový spínač pro vstupy E1 až E2
7.	Indikační kontrolky pro vizualizaci výstupů A1 a A2

4.1.3 KUKA smartPAD

Ovládací panel KUKA smartPAD umožňuje řídit všechny KUKA roboty a taky všechny KUKA Sunrise Cabinets v daném jazyce. KUKA smartPAD má konektor, který lze zapojit za chodu robotické buňky, proto je jednoduché ho kdykoliv zapojit nebo odpojit. Panel obsahuje osm pohybových kláves, jimiž lze ovládat až osm os robotu. Taky lze robotu ovládat pomocí 6D myši, která nabízí možnost změny orientace robotu ve třech nebo ve všech stupních volnosti. Načtení programu je možné provést díky dvěma snadno přístupným USB přípojkám, které jsou schovány pod gumovými kryty. Pomocí KUKA smartPAD lze vytvořit program robotu pomocí inline formulářů. Na obrázku 4.4 a 4.5 je možné vidět smartPAD s popisky, jež jsou označeny čísly. Každé číslo má svou vysvětlivku, která je v tabulce 4.2 popsána. [22]

Tabulka 4.2: Popis k jednotlivým prvkům KUKA smartPAD [22]

Číslo popisku	Popis
1.	2 konektory USB-2.0 s krytem
2.	Tlačítko pro odpojení smartPAD
3.	Přepínač druhů provozu
4.	Tlačítko NOUZOVÉHO ZASTAVENÍ
5.	Space Mouse (6D myš): K ručnímu ovládaní robotu
6.	Pohybové klávesy: K ručnímu ovládaní robotu
7.	Ruční poutka se suchým zipem
8.	Tlačítko k nastavení override programu
9.	Tlačítko k nastavení ručního override
10.	Připojovací kabel
11.	Stavová tlačítka: Stavová tlačítka slouží hlavně k nastavení parametrů z volitelných balíčků.
12.	Tlačítko Start: Tlačítkem Start se spouští program.
13.	Tlačítko Start - Zpět: Tlačítkem Start - Zpět se spouští program směrem zpět. Program se zpracovává v krocích.
14.	Tlačítko STOP: Tlačítkem STOP se zastaví probíhající program
15.	Klávesa klávesnice: Zobrazuje klávesnici
16.	Klávesa Hlavní menu Tlačítko Hlavní menu zobrazí nebo skryje hlavní menu na smartHMI.

4.1.4 Siemens HMI panel

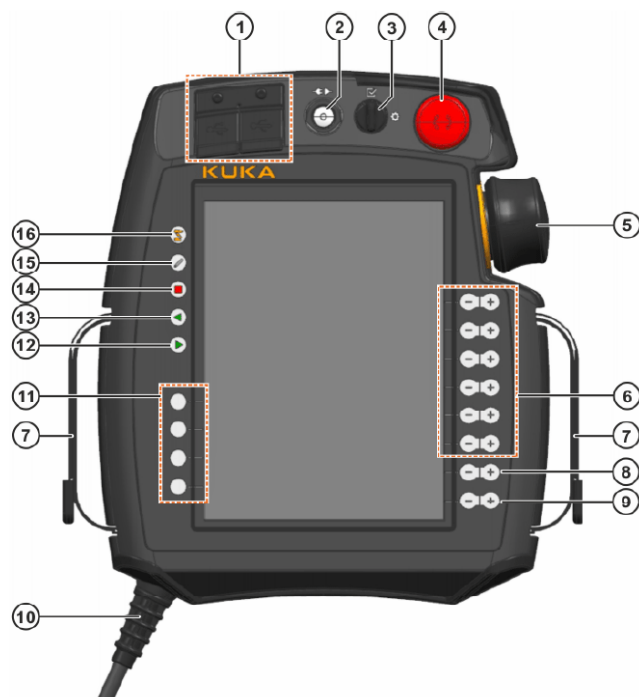
HMI panel je umístěn v přední části robotické buňky a jedná se o SIMATIC panel KTP900 Basic PN. Tento panel je na obrázku 4.6, kde je zobrazen s ilustrační vizualizací. Jedná se o 9 palcový dotykový barevný displej, který má rozhraní Profinetu a 8 funkčních tlačítek. Rozlišení displeje je 800 x 480 px. Panel je napájen stejnosměrným 24 V napětím. [22]

4.1.5 Systém chapadla robotu

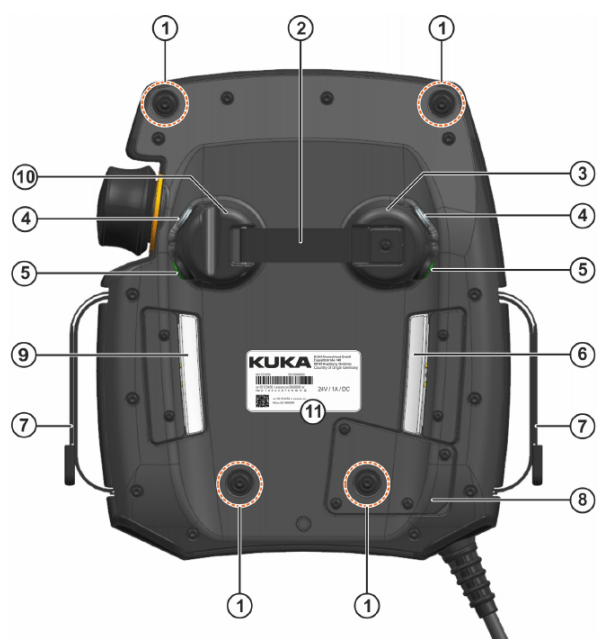
Robot je vybaven dvoučelistovým paralelním chapadlem, které slouží k uchopování, polohování a odkládání dílů. Na obrázku 4.7 je zobrazeno chapadlo robotu s čísly, které mají svůj popis. Popisky jednotlivých prvků jsou k nalezení v tabulce 4.3. Chapadlo funguje na pneumatickém principu. Součástí čelistí jsou dva válcové výstupky, které slouží k jištění uchopeného předmětu. Válcové výstupky přesně zapadají do válcových drážek uchopovaných předmětů z kapitoly 3.1.8. [22]

4.1.6 Montážní deska pneumatiky a elektřiky

Za pomoci jednotky údržby je pneumatické chapadlo robotu zásobováno stlačeným vzduchem, u kterého lze měnit pomocí regulačního ventilu tlak stlačeného vzduchu chapadla. Uzavírací ventil,



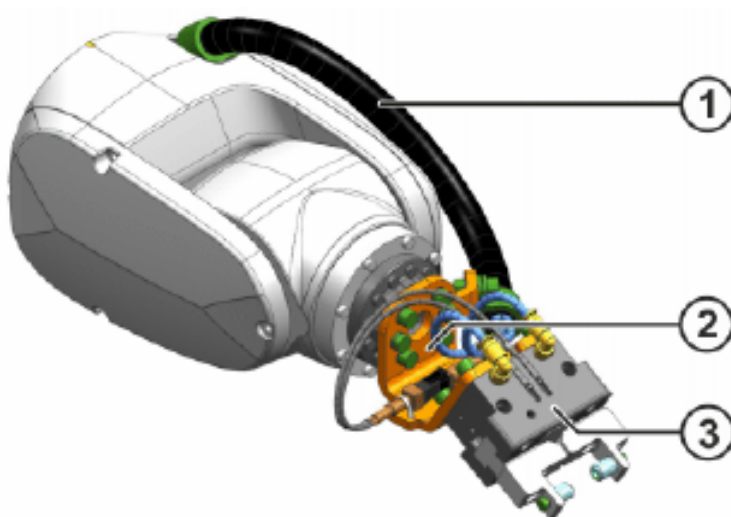
Obrázek 4.4: KUKA smartPAD přední strana
[22]



Obrázek 4.5: KUKA smartPAD zadní strana
[22]



Obrázek 4.6: SIMATIC HMI panel KTP 900 Basic
[22]



Obrázek 4.7: Chapadlo robotu
[22]

jenž se nachází na obrázku 4.8 a jenž je popsán v tabulce 4.4, slouží k zajištění nebo odjištění jednotky údržby. Montážní deska elektriky je stejně jako pneumatika umístěna v rozvodné části robotické buňky, skládá se ze čtyř prvků, které jsou popsány v tabulce 4.5 a na obrázku 4.9 jsou zobrazeny. [22]

4.1.7 PLC systém

Řídicí PLC systém je umístěn v rozvaděči robotické buňky v zadní části vespod. PLC systém je distribuovaný typ řídicího systému, který kombinuje kompaktnost a flexibilitu. Je vhodný zejména v rozsahu středního výkonu pro stroje s distribuovanou inteligencí nebo také stroji sériovými. Procesor tohoto PLC má stejnou funkci jako CPU 1511/ CPU 1512 řady PLC systému S7-1500, proto

Tabulka 4.3: Popis chapadla robotu [22]

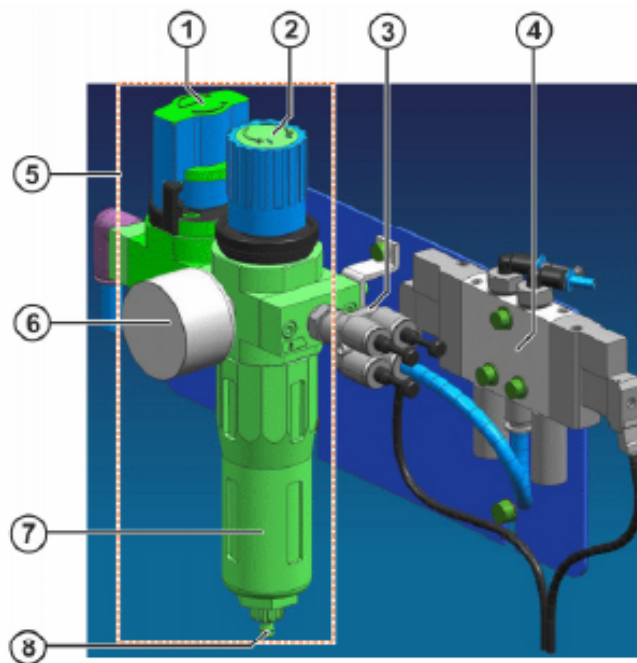
Číslo zařízení	Popis
1.	Přívod energie
2.	Držák chapadla
3.	Dvě paralelní čelisti

Tabulka 4.4: Popis montážní desky pneumatiky [22]

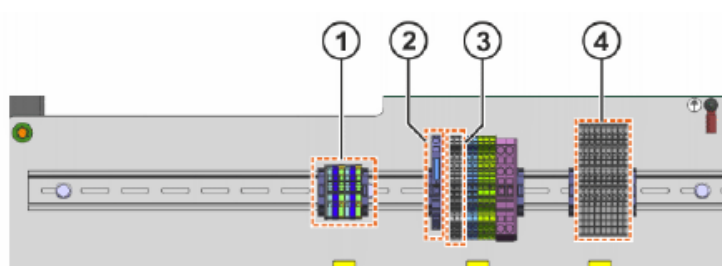
Číslo zařízení	Popis
1.	Uzavírací ventil
2.	Regulační ventil
3.	Vícenásobný rozbočovač
4.	Elektromagnetický ventil
5.	Jednotka údržby
6.	Manometr
7.	Filtrační jednotka
8.	Šroub na upuštění kondenzátu

Tabulka 4.5: Popis montážní desky elektriky [22]

Číslo zařízení	Popis
1.	X0 - Napájení 200-240V, 50-60Hz
2.	F1.5.1 - Pojistka 4A
3.	X1 - Napájení 24V, přípojky PE
4.	X1B - Přípojky Chapač Tlakový ventil, NOUZOVÉ ZASTAVENÍ, potvrďte ochranu obsluhy, bezpečnostní spínač, Profinet



Obrázek 4.8: Montážní deska pneumatiky
[22]



Obrázek 4.9: Montážní deska elektriky
[22]

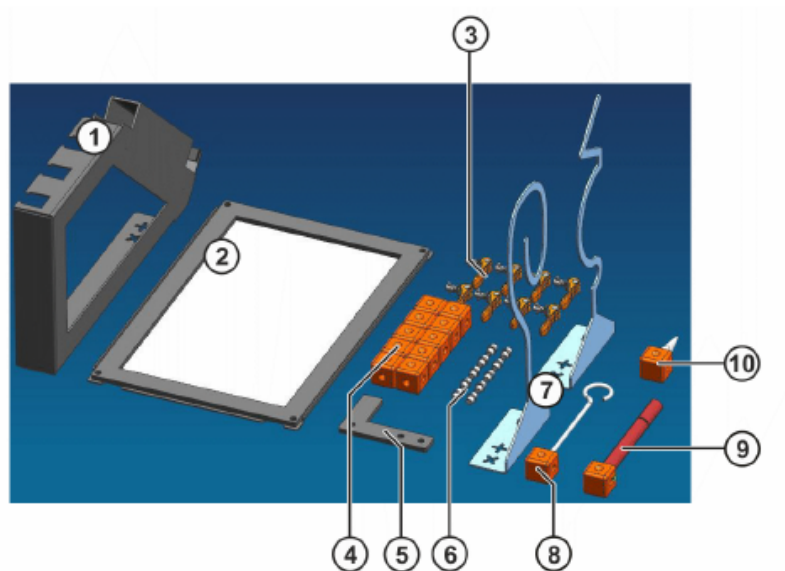
se dá využít pro ovládání KUKA robotu pomocí knihoven mxAutomation. Zařízení je vybaveno ethernetem, jenž komunikuje po Profinetu. PLC je možno dovybavit různými I/O moduly. [22]

4.1.8 Výukové prvky pro realizaci robotických úloh

Robotická buňka obsahuje výukové prvky, což jsou nástroje, pomocí kterých lze realizovat různé úlohy, jako je například překládání kostek, malování pomocí barevné tužky nebo měření pomocí hrotu. Na obr.4.11 lze tyto nástroje spatřit, jednotlivé prvky jsou následně popsány v tabulce 4.6, rovněž je zmíněn počet jednotlivých prvků. [22]



Obrázek 4.10: PLC SIMATIC ET 200SP
[22]



Obrázek 4.11: Výukové přípravky robotického pracoviště KUKA ready2 educate
[22]

4.1.9 Robotické rameno KR 3 R540

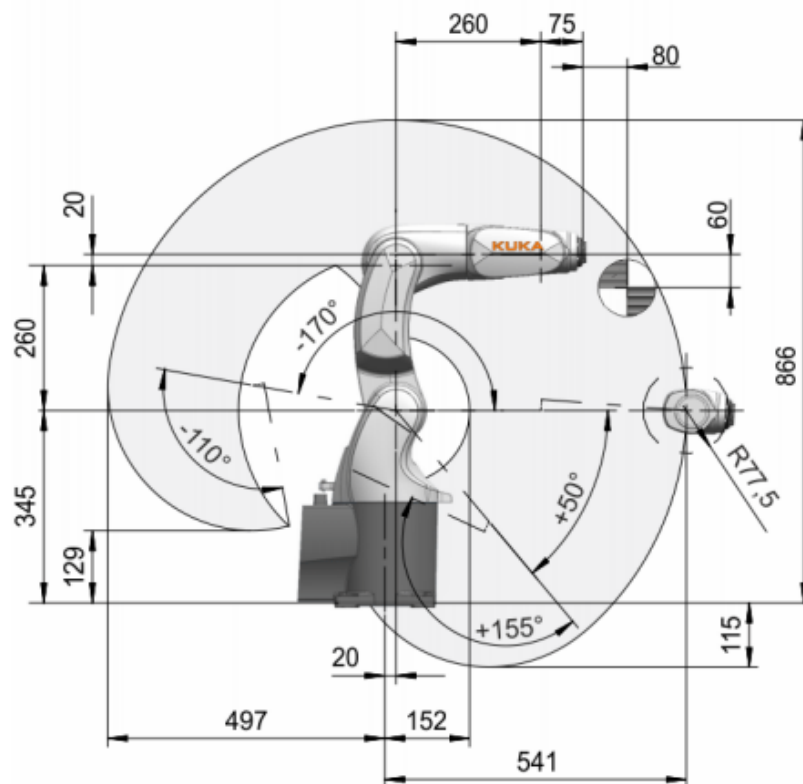
Velkou výhodou robotického ramene KUKA KR 3 AGILUS je to, že se hodí pro prostorově stísněné aplikace a má kompaktní velikost. Jednou z těchto aplikací je například šroubování, balení nebo manipulace s malými díly. Díky své maximální produktivitě na čtvereční metr je tento robot zajímavý pro elektronické obory. Robot je schopen manipulovat s hmotností 2 kg. Přehled parametrů robotu je zaznamenán v tabulce 4.7, na obrázku 4.12 je zobrazen robot a jeho pracovní zóny. [22]

Tabulka 4.6: Popis aplikačních prvků [22]

Číslo zařízení	Popis
1.	Zásobník (1x)
2.	Držák listu (1x)
3.	Excentrická páka (8x)
4.	Kostka 25x25 mm (10x)
5.	Úhelník (1x)
6.	Regálový čep (10x)
7.	Horký drát (2x)
8.	Kostka s háčkem (1x)
9.	Kostka s barevnou tužkou (1x)
10.	Kostka se měřicím hrotem (1x)

Tabulka 4.7: Popis důležitých parametrů KUKA robotu [22]

Důležité technické údaje	
Počet os	6
Hmotnost	26,5 Kg
Maximální dosah	541 mm
Maximální nosnost	3 Kg
Přesnost polohování	$\pm 0,02$ mm
Instalační plocha	179 x 179 mm
Krytí	IP40
Rozsahy os	
A1	$\pm 170^\circ$
A2	$-170^\circ / 50^\circ$
A3	$-110^\circ / 155^\circ$
A4	$\pm 175^\circ$
A5	$\pm 120^\circ$
A6	$\pm 350^\circ$



Obrázek 4.12: Rozsah pracovní zóny robotu KUKA KR 3 R540
[22]

4.2 Definování robotické úlohy

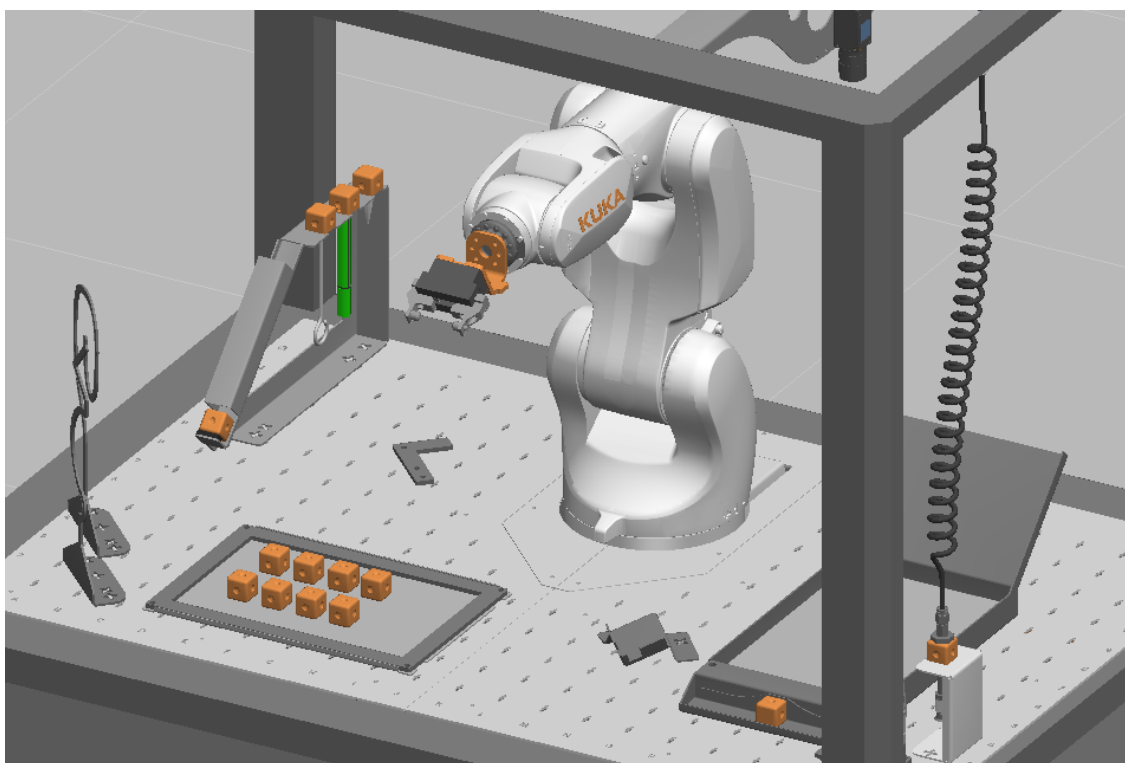
Robotická úloha je definována několika parametry. Jedním z těchto parametrů je realizace řízení robotu pomocí PLC SIMATIC, kdy celý program a jeho vykonávání přebírá právě PLC. Pro realizaci řídicího programu bude použita knihovna KUKA.PLC mxAutomation, která obsahuje důležité funkční bloky pro navázání komunikace a řízení pohybu robotu. Robot bude mít za úkol vykonávat skupinu operací, při jejichž realizaci budou použity přípravky, o nichž se píše v kapitole 4.1.8.

4.2.1 Podrobnější definování robotické úlohy

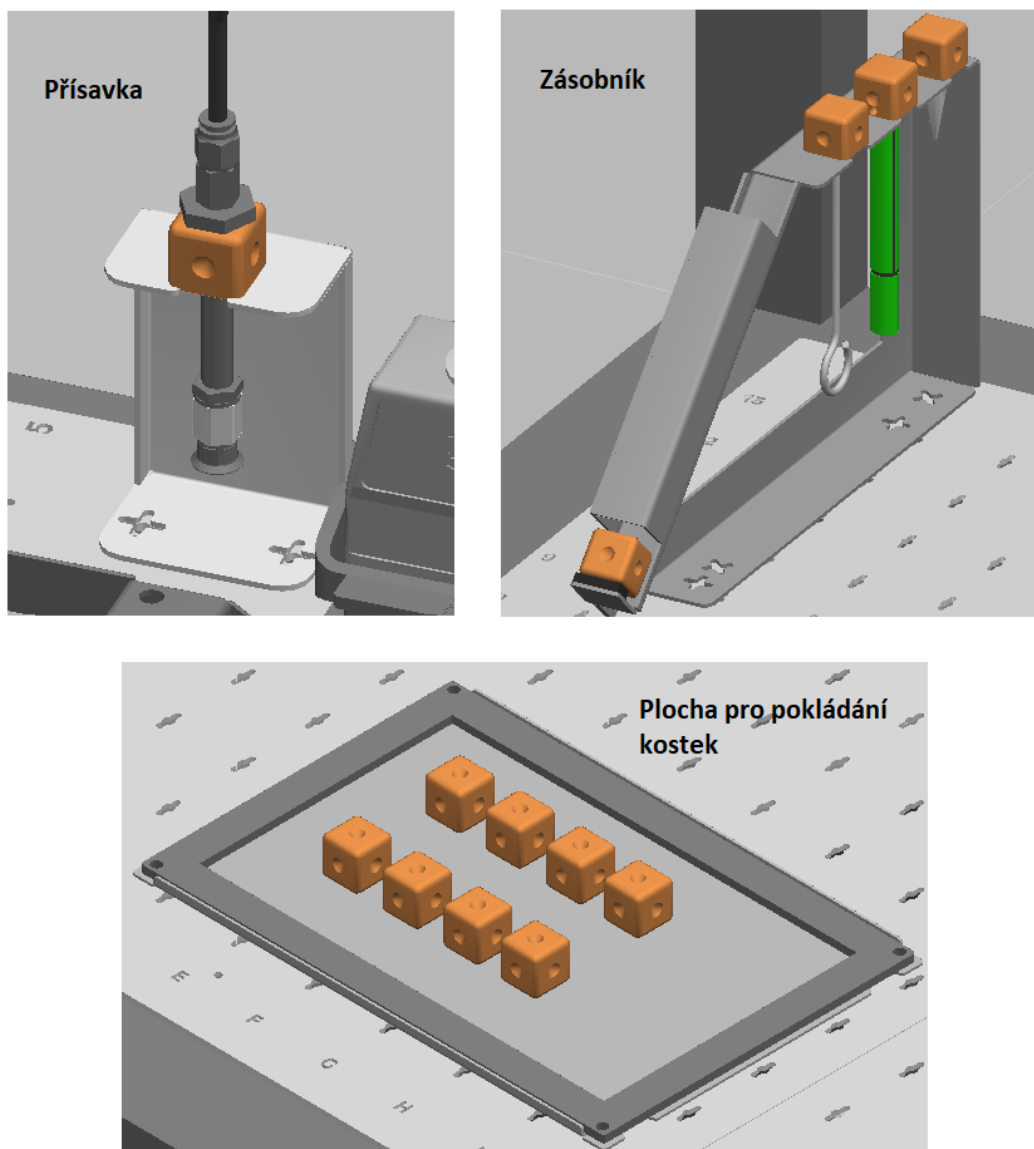
Jelikož je robot vložen do komplexní výukové buňky, viz obr. 4.13, která je vybavena velkým množstvím různých prvků, které mají sloužit k tvorbě různých robotických aplikací. Po důkladném uvážení bylo rozhodnuto, že tyto prvky se využijí při tvorbě robotické úlohy, kdy řízení robotu přebírá PLC systém, jehož program je vytvořen na základech funkcí a funkčních bloků z knihovny mxAutomation. Při vykonávání programu budou použité prvky zobrazeny na obr. 4.14, kde je zobrazena přísavka, zásobník s kostkami i plocha pro položení kostek s ohraničujícím rámem. Kostky budou

vyskládány ve dvou řadách za sebou tak, jak je to zobrazeno na obr. 4.14, kde je toto uspořádání nastíněno. Návrh programu je popsán v jednotlivých bodech označených čísly níže a taky pomocí aktivitního diagramu na obr. B.2.

1. Uchopení přísavky chapačem robotu - Při startu programu se robot přemístí z Home pozice do pozice, při které robot bude ideálně natočen tak, aby mohlo dojít k správnému uchopení přísavky chapačem. Robot následně přísavku přemístí mimo držák.
2. Přesun robotu k zásobníku s kostkami - Jakmile robot bezpečně odebere z držáku přísavku, přemístí se na druhou stranu buňky, kde je umístěn zásobník s kostkami a také volná plocha pro vyskládání kostek.
3. Vyskládání kostek robotem na určité pozice - Robot se natočí tak, aby pod úhlem 90° odebral ze zásobníku postupně 8 kostek a definovaným způsobem tyto kostky vyskládá ve dvou řadách za sebou.
4. Přemístění kostek z daných pozic robotem zpět do zásobníku - Robot se přemístí nad první vyloženou kostku, kterou nasaje a přemístí nad zásobník pod úhlem 90° , kde tuto kostku upustí. Tímto způsobem postupně robot vrátí zpět všech 8 kostek.
5. Vrácení přísavky robotem zpět do zásobníku - Jakmile robot vrátí všechny kostky zpět do zásobníku, natočí se tak, aby mohl přísavku zpět přemístit k zásobníku. Poté se robot přemístí nad držák a lineárním pohybem přísavku položí na držák, otevře chapač, pak se přemístí mimo zásobník.
6. Přesun robotu do Home pozice - Když robot provede celý proces vyskládání a naskladnění kostek a vrátí přísavku, může se přesunout zpět do Home pozice. V tuto chvíli se může program spustit znovu.
7. Podmínky pro správný chod programu - Přísavka musí být uložena v držáku, pokud tak nebude učiněno, robot vykoná bezpečně všechny pohyby, ale nedojde k očekávanému přemístění kostek na určité pozice a zpětnému naskladnění. V zásobníku na kostky musí být umístěno 8 kostek, pokud bude těchto kostek méně, nedojde k vyskládání všech kostek programem definovaných. Také nesmí být v žádném případě v zásobníku více kostek, protože robot s přísátou kostkou může narazit do již uložené kostky v zásobníku.



Obrázek 4.13: Výuková robotická buňka



Obrázek 4.14: Prvky robotické buňky použité pro vytvoření robotické aplikace

Kapitola 5

Návrh a realizace řídicího programu pro průmyslového robotu

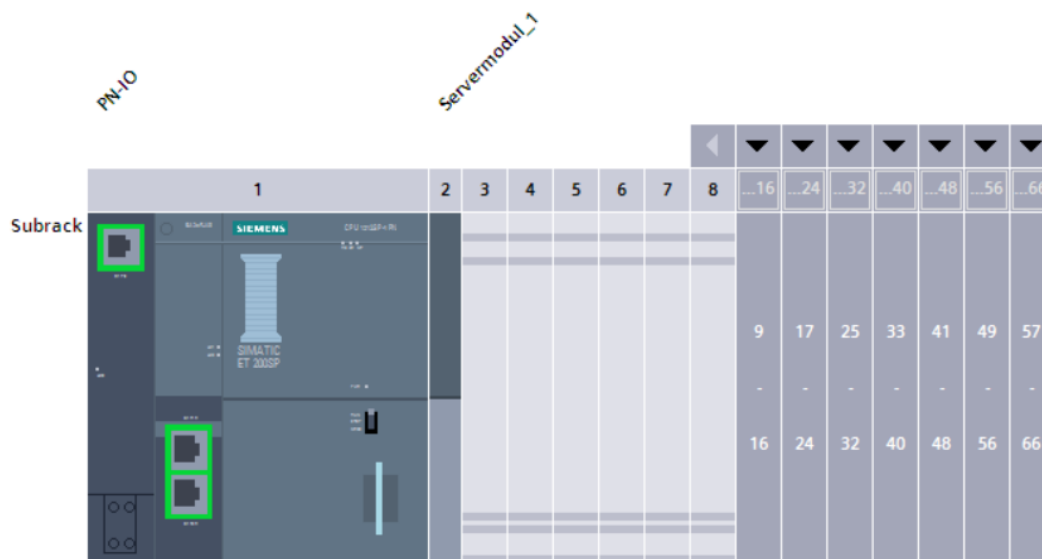
V této kapitole je popsána tvorba softwarového vybavení celé úlohy od nastavení HW konfigurace po tvorbu aplikačního softwaru. Při tvorbě aplikačního programu jsou popsány všechny náležitosti, které jsou potřebné k tomu, aby program komunikoval s řídicí jednotkou robotu. Je potřeba říct, že program, který je popsán v této kapitole, je vytvořen na platformě mxAutomation. Aby komunikace fungovala, musí být toto rozhraní nainstalované v řídicí jednotce robotu.

5.1 Tvorba hardwarové konfigurace

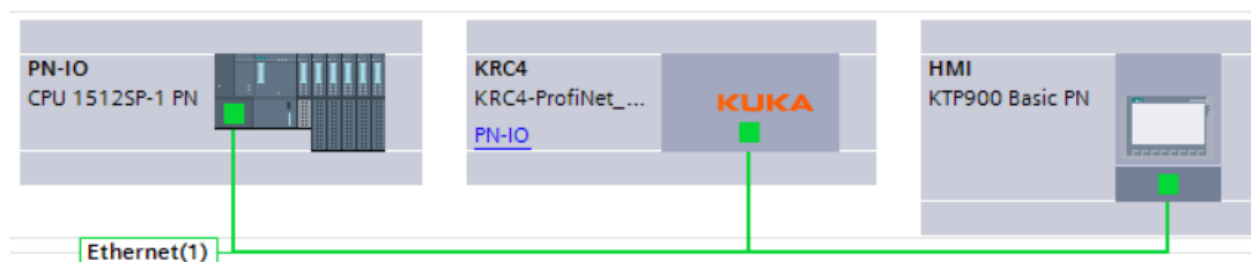
Proto, aby bylo možné vytvořit řídicí aplikaci pro robotu, je nutné nejdříve vytvořit HW konfiguraci, která obsahuje PLC systém, řídicí jednotku robotu a HMI panel. Tato konfigurace je zaznamenána na obr. 5.2, kde jsou všechna (tři) zařízení propojena pomocí Profinetu. Na obrázku 5.1 je zobrazeno použité PLC a jeho samotná konfigurace, která se skládá z jednoho Servermodulu.

5.2 Členění programu

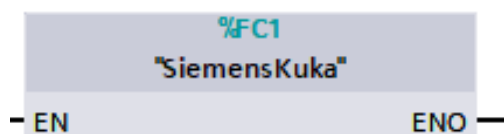
Program se skládá z dvou druhů bloků, a to z těch, které obstarávají komunikaci s robotem, a pak těch, které slouží k tvorbě aplikačního programu, díky nimž robot vykonává zadané operace. Všechny tyto bloky jsou umístěny ve funkci s názvem „SiemensKuka“ v organizačním bloku 1, viz obr. 5.3. Dále projekt obsahuje ještě funkční bloky pro ovládání robotu, pak i ty, díky nimž vykoná nějaký pohyb automatizovaným způsobem. Program je souhrnně vytvořen z mnoha funkčních bloků a funkcí z knihovny mxAutomation, avšak kromě toho taky obsahuje nové funkční bloky, které řeší a ošetřují různé programové stavy.



Obrázek 5.1: HW konfigurace PLC SIMATIC ET 200SP



Obrázek 5.2: Sítová HW konfigurace celého projektu

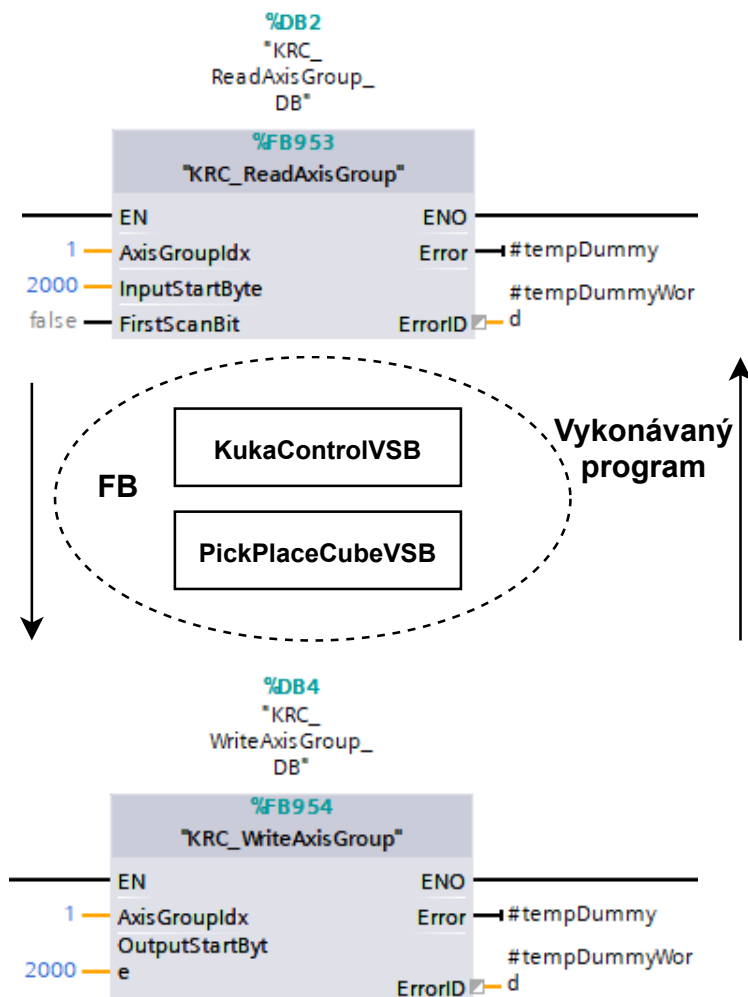


Obrázek 5.3: Hlavní blok, který obsahuje celý program

5.3 Hlavní systémové bloky

V této kapitole jsou rozebrány funkční bloky, které jsou potřebné ke spuštění a navázání komunikace mezi robotem a PLC řídicím systémem. Mezi tyto bloky patří „KRC_ReadAxisGroup“ a „KRC_WriteAxisGroup“, viz obr. 5.4 i s jejich umístěním v rámci programu. Tyto funkční bloky slouží pro čtení a zápis dat do programu, který je mezi ně vložen. Další funkční bloky, které jsou potřebné pro správnou funkci komunikace s použitím rozhraní mxAutomation, jsou umís-

těny ve funkčním bloku „KukaControlVSB“. Jedná se například o funkční bloky „KRC_Initialize“, „KRC_AutomaticExternal“, „KRC_AutoStart“, které mají na starosti navazování a řízení komunikace mezi robotem a PLC.

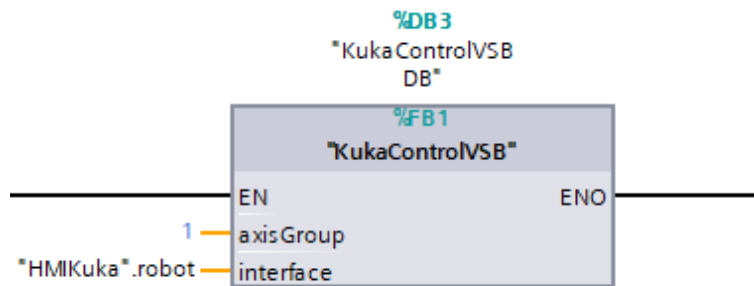


Obrázek 5.4: Funkční bloky pro čtení a zápis dat do robotické jednotky

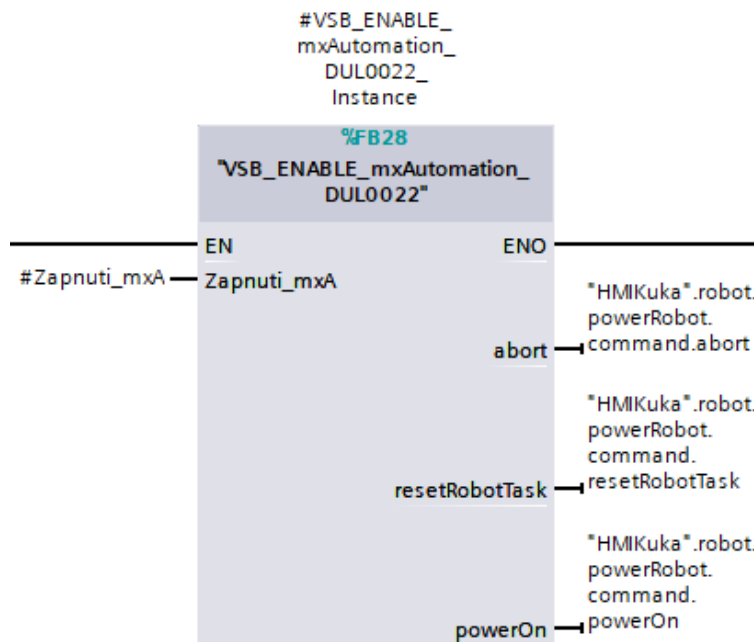
5.4 Funkční bloky pro ovládaní robotu

Tyto bloky jsou obsaženy ve funkčním bloku „KukaControlVSB“, pro představu viz obr. 5.5, tento blok hlavně obsahuje komponenty, které umožňují pracovat s robotem, ovládat jeho nástroje, znamenávat a ukládat pozice robotu. Rovněž je nezbytné zmínit, že v tomto bloku jsou použity funkční bloky, které jsou důležité pro správnou funkčnost komunikace mezi robotem a PLC. Jedním

z nejdůležitějších funkčních bloků je „VSB_ENABLE_mxAutomation_DUL0022“, který je možno pozorovat na obr. 5.6, tento blok není součástí mxAutomation a je nutné si ho doprogramovat.



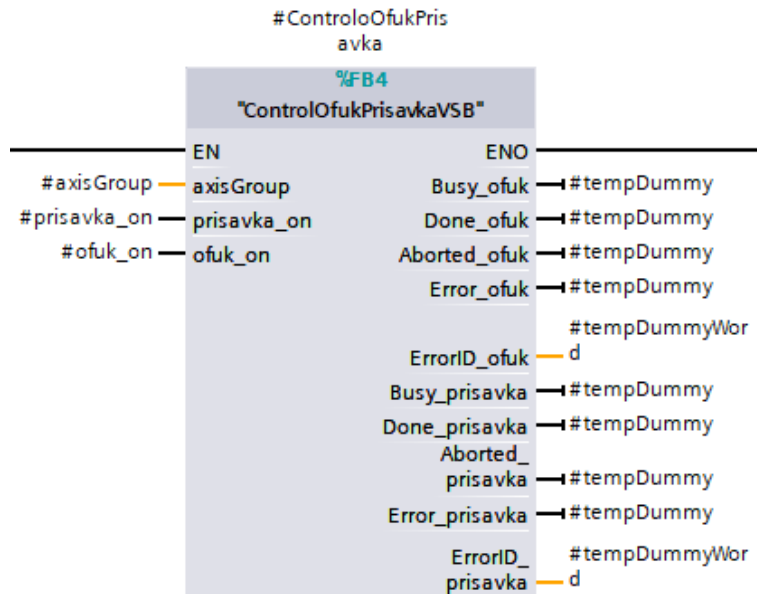
Obrázek 5.5: Funkční blok KukuControlVSB



Obrázek 5.6: Funkční blok pro zapnutí rozhraní mxAutomation

5.4.1 Funkční blok ControlOfukPrisavkaVSB

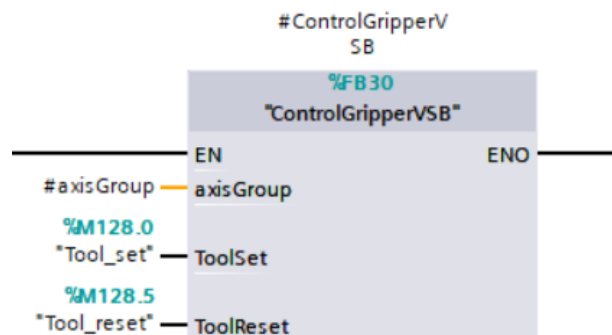
Tento programový blok (viz obrázek 5.7), jenž má na starosti ovládání přísavky a ofuku, což je využito hlavně při nasávání kostek a jiných výukových předmětů, ovládá spouštění a vypínání jednotlivých funkcionalit pomocí vzestupné hrany přivedené na vstupy „prisavka_on“ a „ofuk_on“.



Obrázek 5.7: Funkční blok pro ovládání ofuku a přísavky

5.4.2 Funkční blok ControlGripperVSB

Ovládání chapadla je realizováno pomocí funkčního bloku, viz obrázek 5.8, kde jsou vidět dva vstupy; „ToolSet“ pro sepnutí chapadla a „ToolReset“ pro otevření chapadla. Oba tyto vstupy vykonají danou operaci na základě přivedení náběžné hrany na jejich vstup.

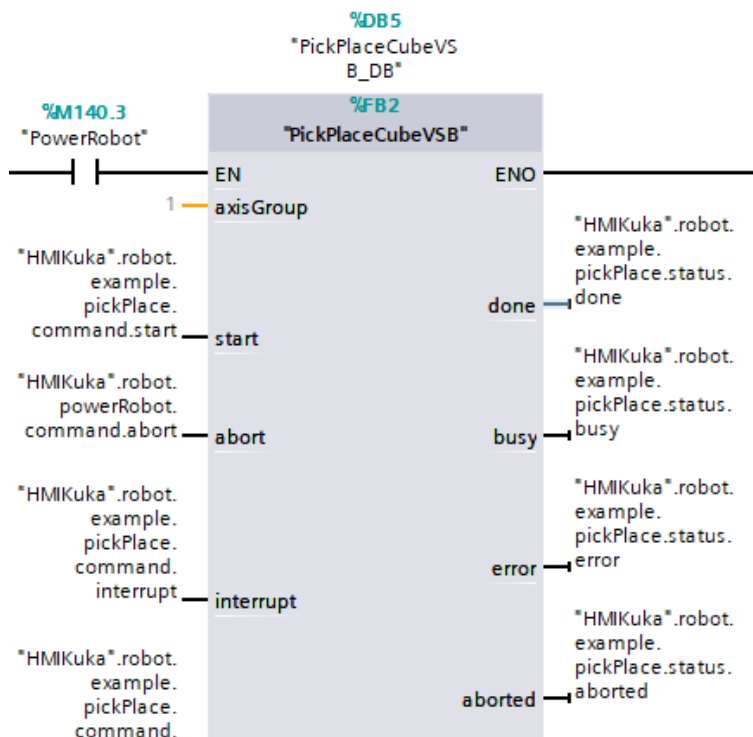


Obrázek 5.8: Funkční blok pro ovládání chapače

5.5 Programové bloky

Všechny funkční bloky, které zabezpečují chod programu, jsou obsaženy ve funkčním bloku, který je graficky znázorněn na obr. 5.9 a jehož název je „PickPlaceCubeVSB“. V tomto programovém

funkčním bloku je obsaženo 52 pozic, do kterých se robot dostane během vykonávání programu. Všechny pozice jsou uloženy v datovém bloku s názvem „MxADBPosition“.



Obrázek 5.9: Funkční blok PickPlaceCubeVSB

5.5.1 Pohybový funkční blok MoveAxisAbsoluteVSB

Pohybový funkční blok „MoveAxisAbsoluteVSB“, viz obr. 5.10, slouží k tomu, aby se robot přemístil z aktuální pozice, kde se zrovna nachází, do požadovaného bodu pomocí natáčení jednotlivých os. Funkční blok má spoustu vstupů a výstupů, které jsou popsány v tabulce 5.1.

5.5.2 Pohybový funkční blok MoveLinearAbsoluteVSB

Tento pohybový funkční blok má za úkol přemístění robotu z pozice, v níž se právě nachází do pozice, která mu byla zadána pomocí pohybu po kartézských osách X, Y, Z, A, B, C. Funkční blok „MoveLinearAbsoluteVSB“ je vidět na obr. 5.11, kde jsou zobrazeny jeho vstupy a výstupy, jež jsou popsány v tabulce 5.2.

5.5.3 Funkční blok ConnectSuctionCup

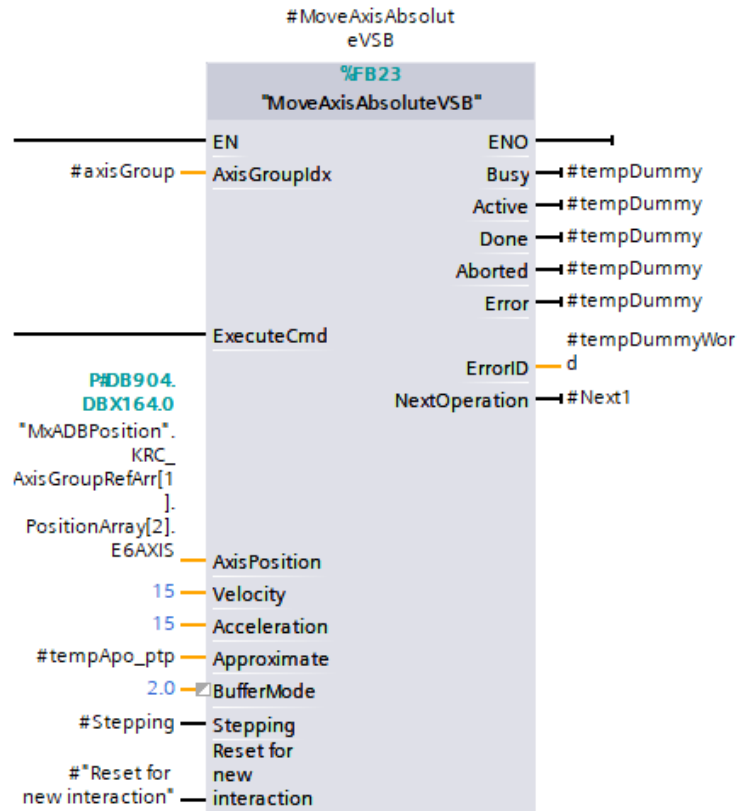
Použití tohoto funkčního bloku je využito pro odebírání 8 kostek. V programu je tento blok použit osmkrát. Cílem tohoto bloku je, aby robot nasál kostku pomocí přísavky, vytáhl ji ze zásobníku,

Tabulka 5.1: Vstupy a výstupy funkčního bloku MoveAxisAbsoluteVSB

Vstupy		
Název	Datový typ	Popis
AxisGroupIdx	INT	Index skupiny os
ExecuteCmd	BOOL	Spuštění pohybu v případě vzestupné hrany signálu.
AxisPosition	E6AXIS	Datová struktura E6AXIS obsahuje hodnoty úhlu nebo hodnoty posunu pro všechny osy skupiny os v koncové poloze.
Velocity	INT	Rychlost0... 100% Odkazuje na maximální hodnotu uvedenou v řídicí jednotce robotu. Výchozí: 0% (= rychlost se nezmění)
Acceleration	INT	Rychlost0... 100% Odkazuje na maximální hodnotu uvedenou v řídicí jednotce robotu. Výchozí: 0% (= rychlost se nezmění)
Approximate	APO	Parametr aproximace
BufferMode	ENUM	Režim, ve kterém je příkaz proveden 1: ABORTING 2: BUFFERED
Stepping	BOOL	Mód pro krokování programu
Reset for new interaction	BOOL	Vstup pro resetování bloku před dalším zavoláním
Výstupy		
Název	Datový typ	Popis
Busy	BOOL	TRUE = výpis byl přenesen
Active	BOOL	TRUE = pohyb se aktuálně provádí
Done	BOOL	TRUE = pohyb se zastavil
Aborted	BOOL	TRUE = příkaz / pohyb byl zrušen
Error	BOOL	TRUE = chyba ve funkčním bloku
ErrorID	INT32	Číslo chyby
NextOperation	BOOL	Informace o dosažení požadovaného bodu

Tabulka 5.2: Vstupy a výstupy funkčního bloku MoveLinearAbsoluteVSB

Vstupy		
Název	Datový typ	Popis
AxisGroupIdx	INT	Index skupiny os
ExecuteCmd	BOOL	Spuštění pohybu v případě vzestupné hranysignál.
Position	E6POS	Souřadnice Kartézské koncové polohy Datová struktura E6POS obsahuje všechny komponentykoncová poloha (= poloha TCP vzhledem k počátku souřadnicového systému BASE).
Velocity	INT	Rychlost0... 100% Odkazuje na maximální hodnotu uvedenou v řídicí jednotce robotu. Výchozí: 0% (= rychlost se nezmění)
Acceleration	INT	Rychlost0... 100% Odkazuje na maximální hodnotu uvedenou v řídicí jednotce robotu.. Výchozí: 0% (= rychlost se nezmění)
CoordinateSystem	COORDSYS	Souřadnicové systémy, ke kterým jsou kartézské souřadnice koncové polohy
OriType	ENUM	Řízení orientace TCP 0: VAR 1: CONSTANT 2: JOINT
Approximate	APO	Parametr aproximace
BufferMode	ENUM	Režim, ve kterém je příkaz proveden 1: ABORTING 2: BUFFERED
Stepping	BOOL	Mód pro krokování programu
Reset for new interaction	BOOL	Vstup pro resetování bloku před dalším zavoláním
Výstupy		
Název	Datový typ	Popis
Busy	BOOL	TRUE = výpis byl přenesen
Active	BOOL	TRUE = pohyb se aktuálně provádí
Done	BOOL	TRUE = pohyb se zastavil
Aborted	BOOL	TRUE = příkaz / pohyb byl zrušen
Error	BOOL	TRUE = chyba ve funkčním bloku
ErrorID	INT32	Číslo chyby
NextOperation	BOOL	Informace o dosažení požadovaného bodu

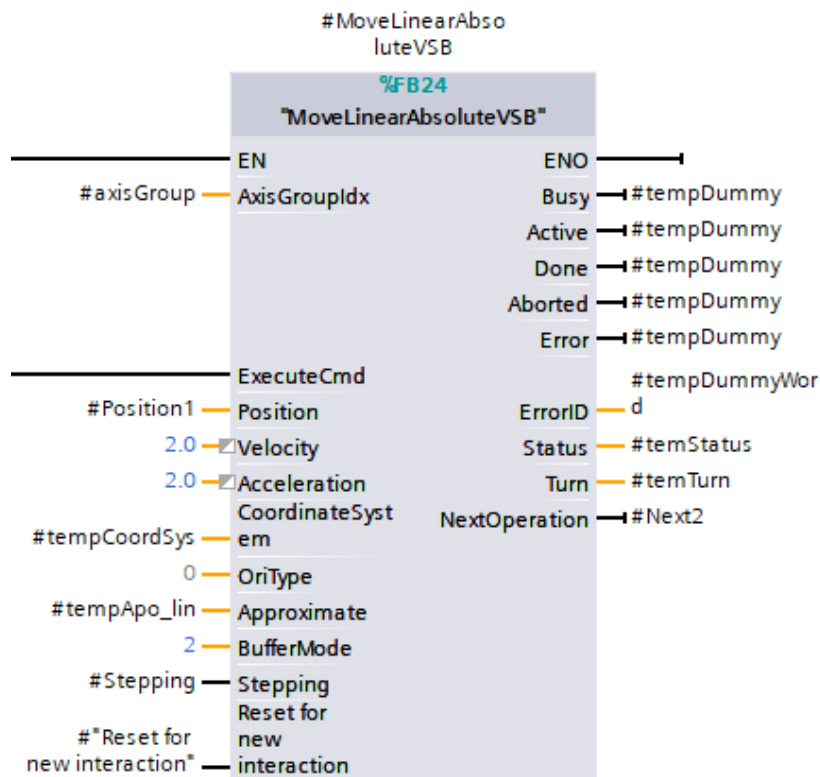


Obrázek 5.10: Funkční blok MoveAxisAbsoluteVSB

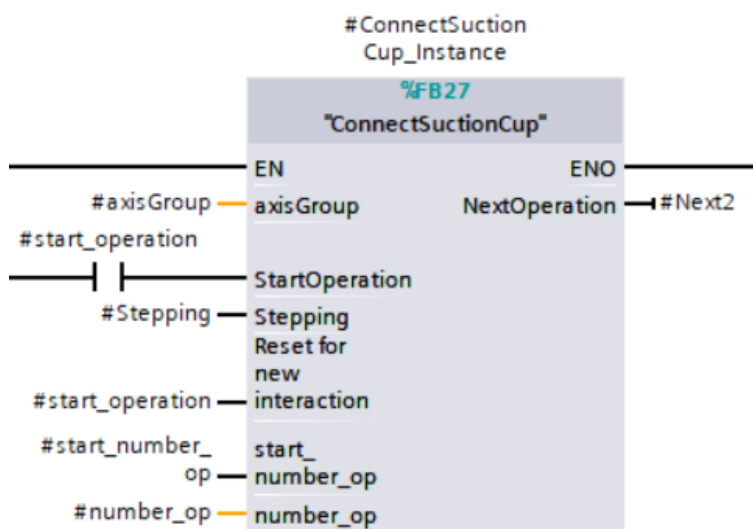
přemístil se nad definovanou pozici, která je situována nad oblast, kde jsou umístěny pozice pro položení kostek. Na obrázku 5.12 je vidět vstup „StartOperation“, kterým se blok spouští, rovněž je zde výstup „NextOperation“, pomocí něhož je poslána informace o ukončení všech pohybů tohoto funkčního bloku.

5.5.4 Funkční blok OperationPickCube

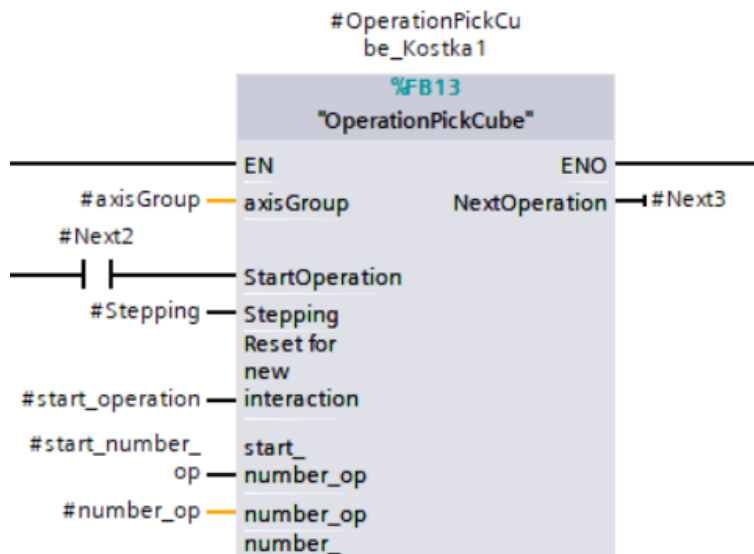
Použití tohoto funkčního bloku je využito pro odebírání 8 kostek. V programu je tento blok použit osmkrát. Cílem tohoto bloku je, aby robot nasál kostku pomocí přísavky, vytáhl ji ze zásobníku, přemístil se nad definovanou pozici, která je situována nad oblast, kde jsou umístěny pozice pro položení kostek. Na obrázku 5.13 je vidět vstup „StartOperation“, kterým se blok spouští, rovněž je zde výstup „NextOperation“, pomocí něhož je poslána informace o ukončení všech pohybů tohoto funkčního bloku.



Obrázek 5.11: Funkční blok MoveLinearAbsoluteVSB



Obrázek 5.12: Funkční blok ConnectSuctionCup



Obrázek 5.13: Funkční blok OperationPickCube

5.5.5 Funkční blok PlaceCube

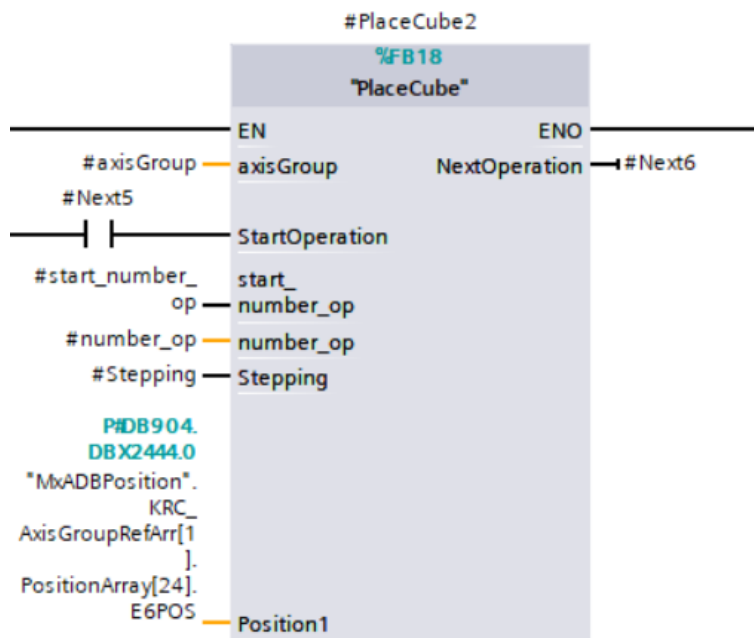
Tento blok zabezpečuje položení kostek na jednotlivé pozice, kdy se robot po spuštění funkčního bloku přemístí nad danou pozicí, kdy následně tuto kostku pomocí kartézského pohybu robotu při změně osy Z položí na definované místo. Použití tohoto bloku v programu odpovídá počtu vytážených kostek, kterých je 8. Po položení se robot opět při změně osy Z přemístí nad položenou kostku. Funkční blok je vidět na obrázku 5.14.

5.5.6 Funkční blok PickCubeFromPosition

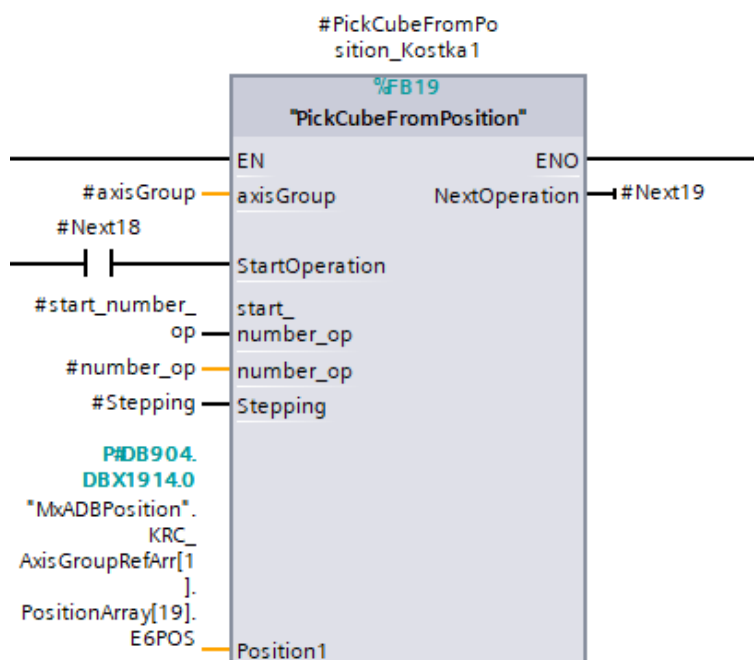
Pro odebírání kostek z pozic je použit funkční blok „PickCubeFromPosition“, který je na obrázku 5.15 vidět se všemi vstupy a výstupy. Funkční blok má za úkol dostat se na určenou pozici kostky, nasát ji a zvednout ji nad podkladovou plochu.

5.5.7 Funkční blok InsertingCubeIntoStorage

Funkční blok „InsertingCubeIntoStorage“, který je zobrazen na obrázku 5.16, slouží k tomu, aby po odebrání kostky z pozice robot tuto kostku přemístil nad zásobník s kostkami. Následně se vypnula přísavka a kostka mohla volně sklouznout v úhlu 90° dolů na spodní část zásobníku, kde při volání dalších instancí tohoto bloku se zásobník plní. Jakmile robot upustí kostku do zásobníku, přemístí se po stejné trajektorii zpět do pozice, v níž je připraven k odebrání kostky zavoláním funkčního bloku „PickCubeFromPosition“.



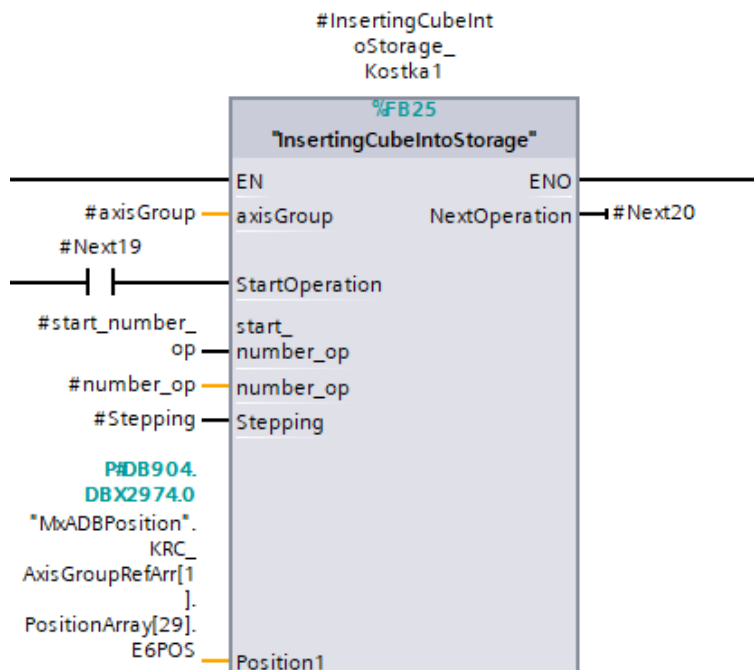
Obrázek 5.14: Funkční blok PlaceCube



Obrázek 5.15: Funkční blok PickCubeFromPosition

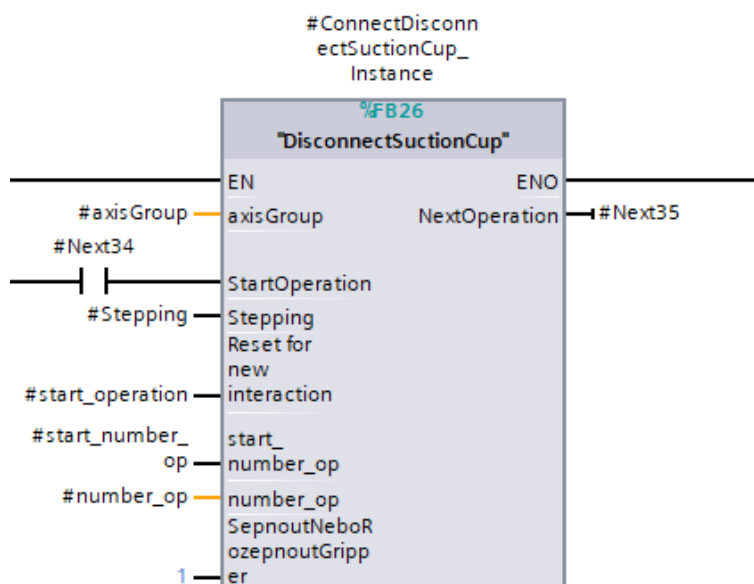
5.5.8 Funkční blok DisconnectSuctionCup

Tento blok je posledním funkčním blokem, jenž je zavolán po vrácení poslední osmé kostky zpět do zásobníku s kostkami. Funkční blok má za úkol po dokončení všech pick and place operací s



Obrázek 5.16: Funkční blok InsertingCubeIntoStorage

kostkami vykonat všechny potřebné operace pro vrácení přísavky zpět držáku. Po vykonání tohoto funkčního bloku se zavolá pohyb, který robotu vrátí do Home pozice. Celý tento funkční blok „DisconnectSuctionCup“ je zachycen na obrázku 5.17.



Obrázek 5.17: Funkční blok DisconnectSuctionCup

Kapitola 6

Realizace uživatelské HMI aplikace

V této kapitole je popsána grafická část aplikace, která je vytvořena pro Siemens HMI panel s označením KTP900 Basic PN. Aplikace se celkově skládá ze 7 oken, přičemž budou popsány pouze tři hlavní okna, která jsou nejdůležitější pro obsluhu a ovládání robotu.

6.1 Přihlašovací okno HMI aplikace

Tato část aplikace je určena pro zapnutí externího řízení založeného na rozhraní mxAutomation. A taky slouží k přihlašování a odhlašování uživatelů. Toto okno má přiřazené funkční tlačítko F1, díky kterému je uživatel schopen se do této části aplikace přesunout. Přihlášení slouží k tomu, aby se uživatel poté mohl přesunout do okna, kde dochází k spouštění a krokování robotické aplikace. Přihlašovací okno je zobrazeno na obrázku 6.1, kde je taky vidět, že grafické rozhraní je koncipováno pomocí šedé barvy.

6.2 Okno HMI aplikace pro ovládání robotu

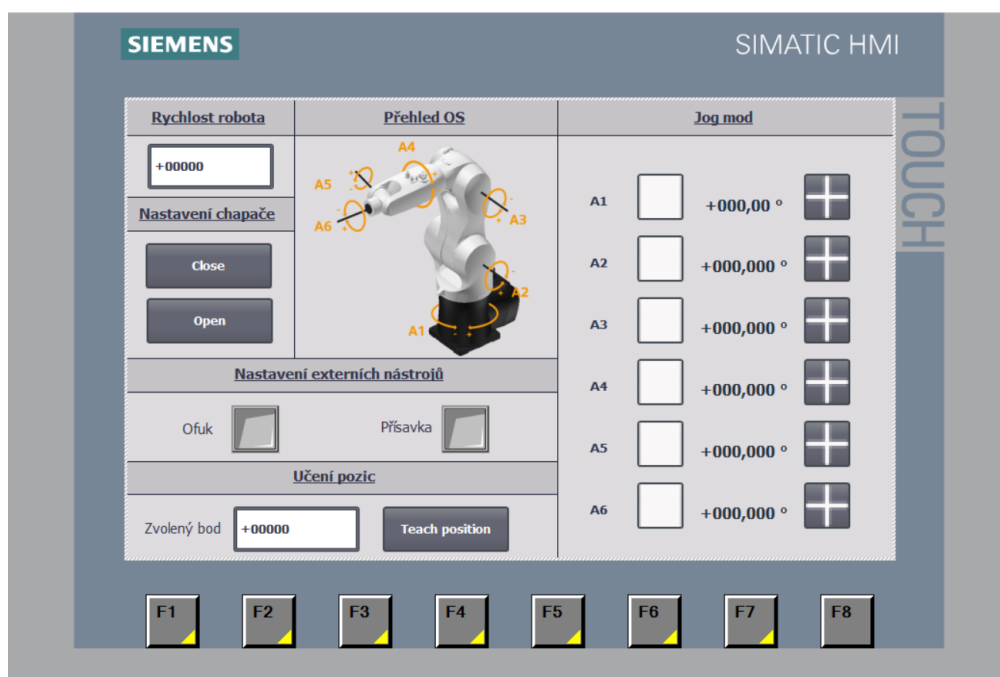
Toto okno slouží k ovládání robotu a jeho nástrojů. Okno je zobrazeno na obrázku 6.2 a jsou zde vidět různé možnosti ovládání robotu od Jog módu, nastavování rychlosti, zavírání a otevírání čelistí chapadla robotu, nastavování externích nástrojů, což je přísavka a ofuk. Poslední možností, která je obsažena v tomto okně, je učení pozic robotu pro tvorbu programu. Okno pro ovládání robotu má přiřazené funkční tlačítko F2, pomocí kterého se uživatel přesune do tohoto grafického rozhraní. Uživatel nemusí být přihlášený, aby se mohl přesunout do tohoto okna.

6.3 Okno HMI aplikace pro spouštění robotické operace

Třetím oknem robotické aplikace je okno, které obsahuje možnost spouštět a krokovat program. Proto, aby se uživatel mohl přesunout do tohoto okna, které má přiřazené funkční tlačítko F3, musí



Obrázek 6.1: Přihlašovací okno HMI aplikace



Obrázek 6.2: Okno HMI aplikace pro ovládání robotu

být uživatel přihlášen jako administrátor. Grafický obraz tohoto okna je vidět na obrázku 6.3, kde je na pravé straně fotka, která zobrazuje reálný průběh aplikace. Na levé straně jsou umístěna tlačítka

pro spouštění aplikace, přerušení a opětovného povolení vykonávání programu. Pokud uživatel chce robotu přesunout do určitého bodu robotické aplikace, musí zvolit číslo této pozice a potvrdit přesun do tohoto bodu tlačítkem „Move position“.

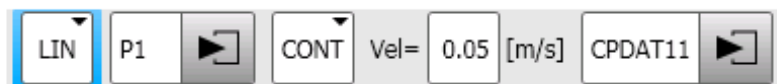


Obrázek 6.3: Okno HMI aplikace pro spouštění robotické operace

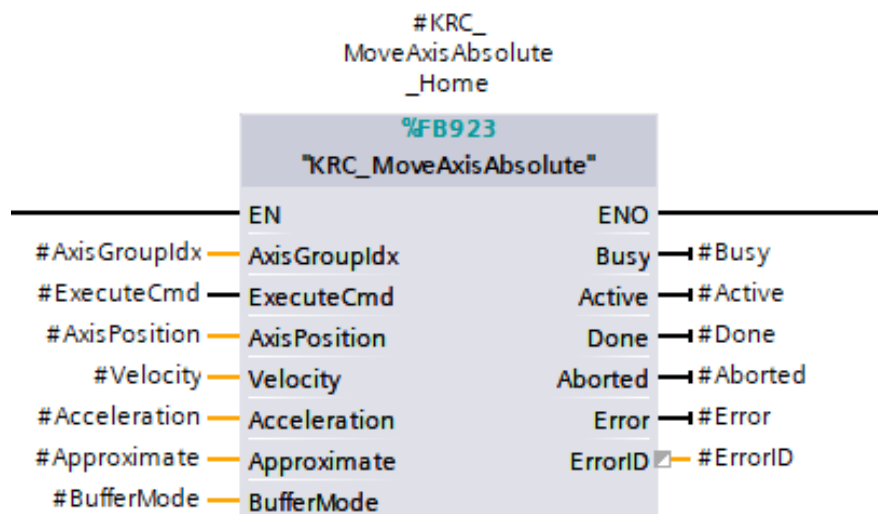
Kapitola 7

Zhodnocení výsledků řešení

V této kapitole je provedeno zhodnocení výsledků a porovnání tvorby programu za pomoci knihovny mxAutomation, která byla aplikována v této práci, oproti klasické metodě založené na inline formulářích. V klasické metodě, která je založena na programovacím jazyce „kuka robot language“, se využívá grafická metoda inline formuláře, viz obrázek 7.1. Výhoda inline formuláře je v tom, že lze rychle v jednom řádku parametrizovat pohyb robotu. Samozřejmě toto lze vztáhnout pouze k tvorbě pohybových příkazů. U inline formuláře je uživatel schopen rychle přidávat další a další pohyb. Naproti tomu program, který je tvořený v LD nebo FB jazyce za pomoci knihovny mxAutomation, má tu nevýhodu, že není tak efektivní z pohledu prostoru kódu. Další nevýhodou u této metody je, že nedochází k sekvenčnímu vykonávání programu, tak jako tomu je u inline formulářů, které se vykonávají řádek po řádku. U funkčních bloků z knihovny mxAutomation, viz obr. 7.2, nevýhoda spočívá v tom, že pokud uživatel kopíruje programové bloky z důvodu zrychlení tvorby programu, může dojít k tomu, že zapomene vytvořit pro funkční blok nový instanční datový blok. Tímto způsobem si uživatel může vytvářet nevědomky chyby v programu, což u inline formuláře nemůže nastat. Největším problémem u pohybových funkčních bloků mxAutomation s označením KRC je to, že tyto funkční bloky nemají výstup, na který by byla zasílána informace o dosažení požadované pozice robotem. Takovou funkcionalitu nabízí funkční blok z knihovny mxAutomation, jenž má označení MC. U funkčních bloků s označením KRC si musí programátor tuto funkcionalitu doprogramovat.



Obrázek 7.1: Inline formulář pro KRL



Obrázek 7.2: Funkční blok knihovny mxAutomation

Kapitola 8

Závěr

Hlavním cílem diplomové práce bylo vytvořit vizualizaci a řídicí program pro průmyslového šestiosého robotu KUKA, který je součástí výukové buňky ready2_educate. Vytvořený řídicí program robotu je vykonáván z PLC zařízení SIMATIC ET 200SP, byl vytvořen za pomoci knihovny funkcí mxAutomation. Jednotlivé programové funkcionality vytvořeného programu jsou spouštěny z HMI panelu SIMATIC KTP900 Basic PN za pomoci vizualizace, která obsahuje tři okna s různými možnostmi ovládání robotu. Pro tvorbu praktické části diplomové práce bylo potřeba analyzovat první dva body zadání, a to jakým způsobem lze realizovat pohyb šestiosého stacionárního robotu a dále analyzovat problematiku knihovny funkcí mxAutomation. Po dokončení analýzy jednotlivých problematik bylo zjištěno, že robotický program může vykonávat pohyby s pomocí pohybových funkcí jako například PTP, LIN. Dále bylo zjištěno, jakým způsobem lze tvořit program pro PLC a jak je tento program zpracováván, jak je posílán do řídicí jednotky robotu, aby došlo k vykonání požadované programové instrukce.

Po studiu teorie byla prozkoumána výuková buňka ready_2educate, bylo zjištěno, jaké má parametry a z jakých prvků se skládá. Po analýze prvků byla definována robotická úloha, ve které byly použity prvky, jako je přísavka, kostky, zásobník. Pro demonstraci řízení robotu pomocí PLC byl vytvořen program, v němž má robot za úkol vzít přísavku z držáku, přemístit se k zásobníku s kostkami a následně vyskládat osm kostek do dvou řad po čtyřech kostkách vedle sebe. Vyskládané kostky poté robot vrátí do zásobníku a přemístí k držáku na přísavku, kterou do tohoto držáku uloží. A nakonec se vrátí zpět do Home pozice.

Při ožiování komunikace mezi PLC a řídicí jednotkou robotu bylo zjištěno, že pro spuštění komunikace je potřeba podle určité posloupnosti poslat na dané proměnné skokově logickou jedničku. Z toho důvodu byl vytvořen funkční blok „VSB_ENABLE_mxAutomation_DUL0022“, který tento proces automatizuje. Díky tomuto funkčnímu bloku stačí komunikace mezi PLC a řídicí jednotkou robotu spustit pomocí zmáčknutí jednoho tlačítka. Dále při tvorbě programu bylo zjištěno, že je potřeba pořídit Siemens mobilní panel, protože není možné provádět jogging pozice robotu z vestavěného HMI panelu, který je umístěn v přední části robotické buňky. Důvodem je to, že pokud je

potřeba přesně naučit pozici robotu, programátor musí vidět na tuto pozici. Mobilní panel bohužel nebylo možné pořídit a integrovat celé řešení z důvodu časové a finanční náročnosti. Tento problém byl vyřešen tím, že externí řízení bylo neustále přepínáno na řízení přes robotickou jednotku, aby se mohl použít KUKA SmartPad.

Pro tvorbu robotického programu byly použity pohybové bloky `KRC_MoveAxisAbsolute` a `KRC_MoveLinearAbsolute`. Neboť bylo zjištěno, že tyto programové bloky neposkytují informaci o dosažení požadované polohy robotem, bylo nutné si vytvořit své vlastní pohybové funkční bloky, které tuto informaci budou poskytovat. Při tvorbě takovýchto funkčních bloků se vycházelo z pohybových funkčních bloků knihovny `mxAutomation`, které mají označení `MC` a jsou homologovány podle normy `PLCopen`. Zároveň je nebylo možné použít při tvorbě programu, protože obsahují speciální nastavení uvnitř projektu. Nově vytvořené funkční bloky jsou popsány v kapitolách 5.5.1 a 5.5.2. Tyto funkční bloky neobsahují nic jiného než již zmíněné KRC pohybové bloky, funkční blok poskytující informaci o aktuální poloze robotu a komparátor dosažené a požadované pozice robotu.

V kapitole 7 je provedeno zhodnocení a porovnání výsledků zvolené metody řízení a tvorby robotického programu s klasickou metodou tvorby programu v jazyce `KRL`.

Řídicí program pro robotické rameno byl v průběhu mnohokrát testován a ošetřován. Výsledkem testování programu je video, které je součástí přílohy.

Literatura

1. *Cellule Ready2Educate: Robotised cell to learn and improve your knowledge of 6-axis robotics*. Carpentras: Erm-automatismes, [b.r.]. Dostupné také z: <https://www.erm-automatismes.com/p462-eng-cellule-ready2-educate.html?g=4>.
2. *Osy robota: Jak ovlivňují jeho využití?* Factoryautomation, 2019. Dostupné také z: <https://factoryautomation.cz/osy-robota-jak-ovlivnuji-jeho-vyuziti/>.
3. *Průmyslové roboty zvyšují svůj podíl na trhu*. Brno: Elektroprumysl.cz, 2016. Dostupné také z: <https://www.elektroprumysl.cz/automatizace/prumyslove-roboty-zvysuji-svuj-podil-na-trhu>.
4. *VITRALAB: Příručka Automatizační a robotická technika*. Košice: Www.sjf.tuke.sk, 2011. Dostupné také z: http://www.sjf.tuke.sk/vitralab/files/vystupy/prirucka%5C_cz%5C_final.pdf.
5. *KUKA: KR QUANTEC*. Stuttgart: KUKA, 2019. Dostupné také z: https://www.kuka.com/-/media/kuka-downloads/imported/9cb8e311bfd744b4b0eab25ca883f6d3/kuka-brochure%5C_kr%5C_quantec%5C_en.pdf?rev=260ac511cb7e4844b25865123ea41ad3.
6. *KUKA: KUKA smartPAD*. Stuttgart: KUKA, 2018. Dostupné také z: <https://www.kuka.com/cs-cz/produkty,-slu%5C%5C%BEby/robotick%5C%C3%5C%A9-syst%5C%C3%5C%A9my/%5C%C5%5C%99%5C%C3%5C%ADdic%5C%C3%5C%AD-syst%5C%C3%5C%A9my-robot%5C%C5%5C%AF/smartpad>.
7. *Vybrané partie z robotiky*. Brno: VUT, 2015. Dostupné také z: <https://docplayer.cz/37783557-Vybrane-partie-z-robotiky.html>.
8. *Uvod do mechatroniky, robotiky a systémů řízení pohybu*. Plzeň: Katedra kybernetiky Západočeská univerzita v Plzni, 2012. Dostupné také z: <http://home.zcu.cz/~msvejda/URM/materialy/Uvod%5C%20do%5C%20mechatroniky.pdf>.
9. *Integrace a modelování systému distribuovaného řízení polohy a průmyslového robota*. Praha, 2012. Dostupné také z: <http://docplayer.cz/5731108-Fakulta-elektrotechnicka-integrace-a-modelovani-systemu-distribuvaneho-rizeni-polohy-a-prumysloveho-robota.html>.

10. *KUKA System Software 8.3: Operating and Programming Instructions for System Integrators*. Stuttgart: Wwww.wtech.com, 2015. Dostupné také z: <http://www.wtech.com.tw/public/download/manual/kuka/krc4/KUKA%5C%20KSS-8.3-Programming-Manual-for-SI.pdf>.
11. *NUMERICKÉ METODY I*. Ostrava: VŠB-TUO, 2011. Dostupné také z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/numericke_metody.pdf.
12. *KUKA System Software 5.5: Operating and Programming Instructions for End Users*. Stuttgart: KUKA, 2008. Dostupné také z: <https://www.hgpaction.com/wp-content/uploads/2017/02/KSS-5.5-Operating-and-Programming-Instructions-for-End-Users.pdf>.
13. *KR C2 / KR C3 Expert Programming: KUKA System Software (KSS)*. Stuttgart: KUKA, 2006. Dostupné také z: <https://www.hgpaction.com/wp-content/uploads/2017/02/Expert-Programming-5.2.pdf>.
14. *KUKA Robotics Library*. München: DIGIMETRIX, [b.r.]. Dostupné také z: <https://www.digimetrix.com/products/kuka/>.
15. *Řízení průmyslových robotů prostřednictvím plc automatů*. Těšín: [Http://udrzbapodniku.cz/](http://udrzbapodniku.cz/), 2011. Dostupné také z: <http://udrzbapodniku.cz/hlavni-menu/artykuly/artykul/article/rizeni-prumyslovych-robotu-prostrednictvim-plc-automatu/>.
16. *KUKA.CNC*. Stuttgart: KUKA, 2012. Dostupné také z: https://www.kuka.com/cs-cz/produkty,-slu%C5%BEby/robotick%C3%A9-syst%C3%A9my/software/aplika%C4%8Dn%C3%AD-software/kuka_cnc.
17. *Simulácia a off-line programovanie priemyselných robotických systémov*. Košice: AT&P journal, 2005. Dostupné také z: <https://www.atpjournals.sk/buxus/docs/atp-2005-02-84.pdf>.
18. *Demystifikace programování robotů*. Český Těšín: Control Engineering Česko, 2019. Dostupné také z: <https://www.vseoprmyslu.cz/robotizace/roboty-software/demystifikace-programovani-robotu.html>.
19. *KUKA.PLC mxAutomation*. Praha: KUKA, 2014. Dostupné také z: <https://www.kuka.com/cs-cz/produkty,-slu%C5%BEby/robotick%C3%A9-syst%C3%A9my/software/prov%C3%A1zan%C3%A9-technologie/kuka-plc-mxautomation>.
20. *SIMATIC Robot Integrator for KUKA - Getting Started*. Berlín: Siemens, 2020. Dostupné také z: <https://support.industry.siemens.com/cs/document/109482123/simatic-robot-integrator-for-kuka-getting-started?dti=0%5C%5C&lc=en-TZ>.
21. *Controlling a KUKA Industrial Robot Using a SIMATIC S7-1500*. Berlín: Siemens, 2016. Dostupné také z: <https://docplayer.net/47895646-Controlling-a-kuka-industrial-robot-using-a-simatic-s7-1500.html>.
22. *Application Module ready2educate Provozní návod*. Augsburg: KUKA, 2020.

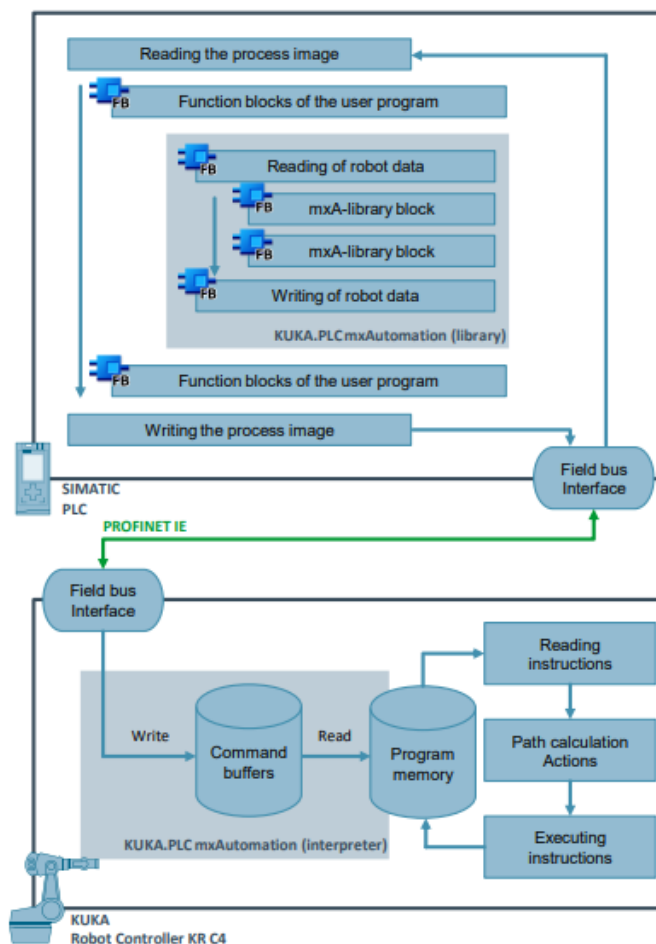
Příloha A

Soubory

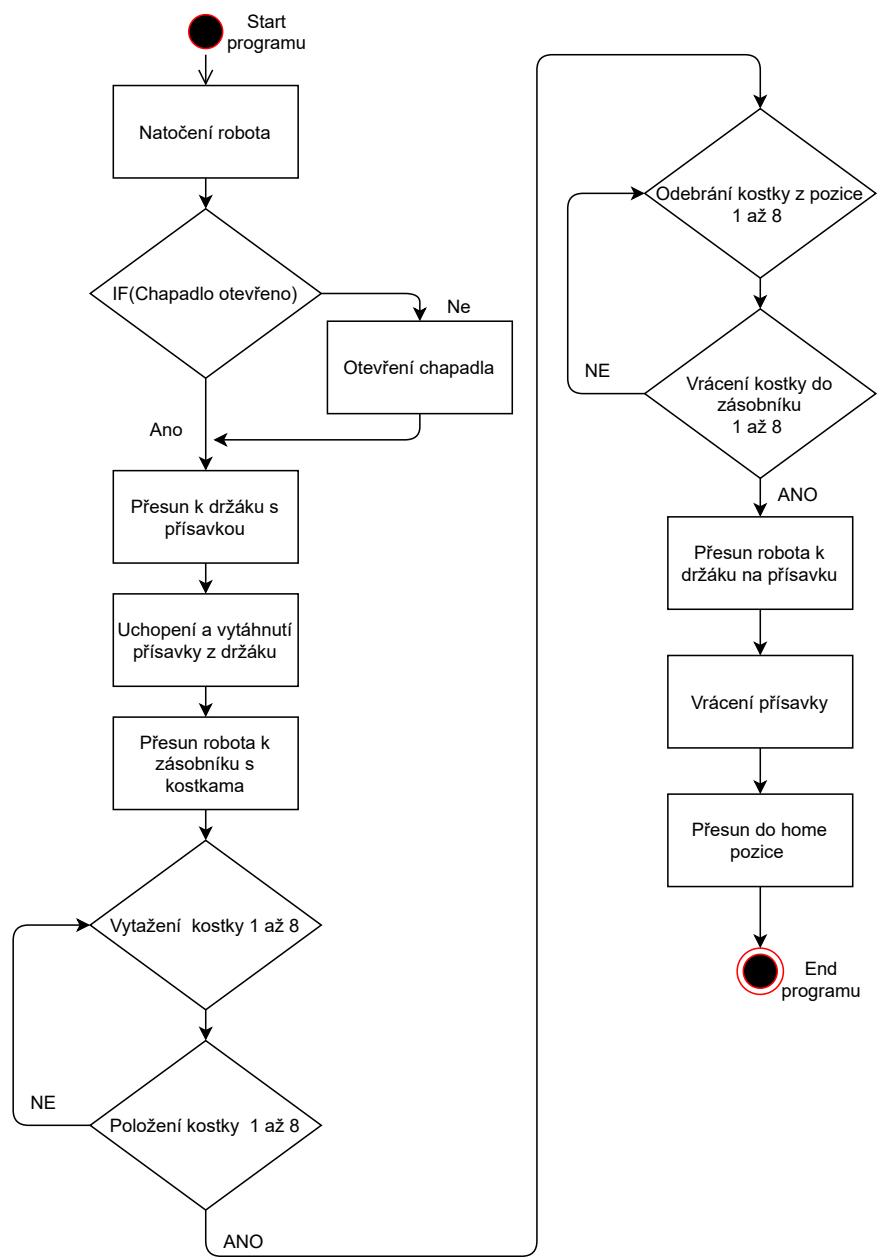
- Sestříhané video z průběhu vytvořené robotické aplikace
- Projekt v TIA portálu
- Dokumentace KUKA Ready2_educate
- Manuál pro přepnutí robota na externí řízení
- Siemens_Library_mxAutomation_30_en

Příloha B

Velké obrázky a tabulky



Obrázek B.1: Diagram zpracování programu
[19]



Obrázek B.2: Aktivitní diagram návrhu programu